0-315-03600-1

National Library
of Canada

Canadian Theses Division

Ottawa, Canada
K1A 0N4

Bibliothèque nationale
du Canada

Division des thèses canadiennes

51495

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

- Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur
ALLAN DAVID HIEBERT

Date of Birth — Date de naissance
JUNE 11, 1958

Country of Birth — Lieu de naissance
CANADA

Permanent Address — Résidence fixe
11308 - 55 AVE
EDMONTON, ALBERTA
CANADA, T6H 0X2

Title of Thesis — Titre de la thèse
COMPUTER SIMULATION OF INSITU, LOW FREQUENCY, ELECTRICAL HEATING OF OIL SAND FORMATIONS

University — Université
UNIVERSITY OF ALBERTA

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée
M.Sc.

Year this degree conferred — Année d'obtention de ce grade
1981

Name of Supervisor — Nom du directeur de thèse
C.E. CAPJACK

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date
JANUARY 23, 1981

Signature
Allan Hiebert

NL-91 (4/77)

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

Ottawa, Canada
K1A 0N4

NL-339 (Rev. 8/80)

THE UNIVERSITY OF ALBERTA

COMPUTER SIMULATION OF IN SITU, LOW FREQUENCY, ELECTRICAL HEATING OF OIL SAND FORMATIONS

by

© ALLAN DAVID HIEBERT

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE

ELECTRICAL ENGINEERING

EDMONTON, ALBERTA
SPRING, 1981

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR ALLAN DAVID HIEBERT

TITLE OF THESIS COMPUTER SIMULATION OF IN SITU, LOW FREQUENCY, ELECTRICAL HEATING OF OIL SAND FORMATIONS

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED SPRING, 1981

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Allan Hiebert

PERMANENT ADDRESS:

11308 - 55 Ave.
Edmonton, Alberta, Canada
T6H 0X2

DATED January 19, 1981

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled "Computer Simulation of In Situ, Low Frequency, Electrical Heating of Oil Sand Formations" submitted by Allan David Hiebert in partial fulfilment of the requirements for the degree of Master of Science.

........C.E. Capjack........

Supervisor

....

........J. Samson........

Date..Jan. 15. - 1981..... F.E. Vermeulen.

Abstract

The mathematical modeling of electrical and thermal phenomena that occur during in situ, low frequency, electrical heating of oil sand, or other materials with similar electrical properties, is studied. Particular attention was directed toward the modeling of the electrical field at an interface between two media of different electrical conductivities. The finite difference method is used to develop a computer program to simulate, in two dimensions, the mathematical model that was developed.

This computer program is used to study the role of various parameters in determining the final temperature profile after a fixed time period and fixed rate of heating. The parameters studied include the position of the electrodes relative to the overlying and underlying formations, the spacing between the electrodes, and the ratio of the conductivity of the oil sand formation to the conductivity of the surrounding formations.

Acknowledgement

I would like to thank my supervisors, Dr. C.E. Capjack, Dr. F.S. Chute and Dr. F.E. Vermeulen, for their very helpful assistance during all stages of my study.

I would also like to thank Dr. M. Cervenan for his help, and J. Fearn and A. Vogan for their efforts in setting up and performing the physical scale model runs that provided data that could be compared to the results from the computer program.

Table of Contents

Chapter Page

1. Introduction 1
1.1 Electrical Preheat In Situ Method 1
1.2 Means of Investigation 3
1.3 Electrical and Thermal Properties of Oil Sand 4
1.4 The Computer Program "MEGAERA" 7
2. The Mathematical Model 11
2.1 Electrical Fields 11
2.1.1 The Quasi-static Approximation 11
2.1.2 Current Continuity Equation 14
2.1.3 Electrical Boundary Conditions 15
2.2 Heat Generation and Tranfer 20
3. Numerical Solution of the Model 23
3.1 Differencing of the Current Continuity Equation ...23
3.2 Differencing the Thermal Equation 28
3.3 Solution of the Difference Equations 29
3.4 Testing of MEGAERA 31
3.4.1 Energy Balance 32
3.4.2 Comparison with Analytic Results 33
3.4.3 Comparison with Physical Models 35
4. Results of MEGAERA Simulation Runs 39
4.1 Effects of the Electrical Conductivity Ratio 39
4.2 Effects of changing the distance between the electrodes 52
5. Conclusions 64
References and Bibliography 67
Appendix I - Listing of the Program MEGAERA 70

Appendix II - Sample Output from MEGAERA 113

List of Tables

Table Page

1.1 Electrical Properties of Reconstituted Oil Sand at 60 Hz and 24° C..5

1.2 Thermal Properties of Reconstituted Oil Sand and Other Materials..5

4.1 Temperature rise and thermal efficency as a function of the relative conductivity of the oil sand and surrounding formations....................43

4.2 Temperature rise and thermal efficency as a function of distance between the electrodes........55

List of Figures

Figure Page

1.1 Typical Vertical Electrode Geometry................9

1.2 Typical Horizontal Electrode Geometry..............9

2.1 Planes of Symmetry in a Five Spot Array............17

3.1 Finite difference grid used in MEGAERA showing the locations at which potential, temperature and conductivities are computed...................25

3.2 Profile of Electrical Potential Across an Conductivity Discontinuity, showing the abrupt change in normal component of the electric field as required for continuity of the normal component of current................................27

3.3 Computer calculated and measured temperatures for the physical model run. The contour plots are computer generated and the measured values are written on top...............................37

4.1 Heating Configuration A............................40

4.2 Heating Configuration B. The thicknesses of these two dimensional geometries are set to one meter for resistance and power calculations........40

4.3 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 10, configuration B, with an electrical conductivity ratio (overburden, underburden/ oil sand) of 10....43

4.4 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 12, configuration B, with an electrical conductivity ratio (overburden, underburden/ oil sand) of 2.....44

4.5 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 18, configuration A, with an electrical conductivity ratio (overburden, underburden/ oil sand) of 2.....45

4.6 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 19, configuration A, with an electrical conductivity ratio (overburden, underburden/ oil sand) of 5.....46

4.7 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 20, configuration B, with an electrical conductivity ratio (overburden, underburden/ oil sand) of 5.....47

4.8 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 21, configuration A, with an electrical conductivity ratio (overburden, underburden/ oil sand) of 1.....48

4.9 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 22, configuration A, with an electrical conductivity ratio (overburden, underburden/ oil sand) of .5....49

4.10 Temperature vs. the ratio of the electrical conductivities of the surrounding formations to that of the oil sand.................................50

4.11 Contour plot of temperature in Celsius at the end of heating for one year at 16 kW/m. Run no. 23, configuration A, with a conductivity ratio (overburden, underburden/ oil sand) of .5..........55

4.12 Contour plot of temperature in Celsius at the end of heating for one year at 16 kW/m. Run no. 24, configuration A, with a conductivity ratio (overburden, underburden/ oil sand) of 1...........56

4.13 Contour plot of temperature in Celsius at the end of heating for one year at 16 kW/m. Run no. 25, overburden half as conductive and underburden twice as conductive as oil sand.......57

4.14 Contour plot of temperature in Celsius at the end of heating for one year at 8 kW/m. Run no. 26, overburden half as conductive and underburden twice as conductive as oil sand.......58

4.15 Contour plot of temperature in Celsius at the end of heating for one year at 12 kW/m. Run no. 32, overburden half as conductive and underburden twice as conductive as oil sand.......59

4.16 Contour plot of temperature in Celsius at the end of heating for one year at 12 kW/m. Run no. 34, configuration A, with a conductivity ratio (overburden, underburden/ oil sand) of .5..........60

4.17 Contour plot of temperature in Celsius at the end of heating for one year at 12 kW/m. Run no. 35, configuration A, with a conductivity ratio (overburden, underburden/ oil sand) of 1...........61

4.18 Temperature midway between the electrodes vs. the distance between the electrodes. Electrode positions are as in configuration A, except for the mixed conductivity ratio where the electrodes were moved up three meters..............62

Nomenclature

A_{ij}	Coefficent of the (i,j) term of the difference equation.
$\bar{B}$	Magnetic flux density vector.
C_{ij}	Coefficent of the (i-1,j) term of the difference equation.
$\bar{D}$	Electrical displacement vector.
$\bar{E}$	Electrical field vector.
E_E	Electrical energy input.
E_H	Total heat generated.
E_T	Thermal energy stored in the formation.
F_{ij}	Coefficent of the (i,j-1) term of the difference equation.
$\bar{H}$	Magnetic field vector.
I^n	Total electrical current during the nth time step.
$\bar{J}$	Electrical current density vector.
M_o	Volumetric heat capacity.
$\dot{Q}$	Instantaneous heating rate.
$\dot{Q}_{AV}$	Average heating rate.
R_{ij}	Coefficent of the (i,j+1) term of the difference equation.
S_{ij}	Coefficent of the (i+1,j) term of the difference equation.
T	Temperature
T_o	Initial temperature of the formation.
V^n	Voltage between the electrodes during the nth time step.

d	Distance between the electrodes.
j	Square root of -1.
k_h	Thermal conductivity.
t	Time.
Δt	Length of time step.
V_{ij}	Volume of the (i,j) grid block.

α	Temperature dependence of electrical conductivity.
ε	Electrical permittivity.
ψ	Electrical potential.
ρ	Electrical charge density.
ρ_s	Electrical surface charge density.
σ	Electrical conductivity.
σ_{24}	Electrical conductivity at 24°C.
ω	Frequency in radians per second.

1. Introduction

1.1 Electrical Preheat In Situ Method

In northern Alberta there exist large deposits of petroleum in the form of oil sand.[1] The oil in these deposits is too viscous to recover by conventional methods. Commercial developments to date have mined the oil sand in open pit mines, then separated out the petroleum (in the form of bitumen) in large extraction plants. The bitumen is upgraded to "synthetic crude", then piped to a refinery.

As the thickness of the earth, or overburden, above the oil sand formation increases, open pit mining becomes uneconomical. Only about ten percent of the oil sand reserves are recoverable by surface mining.[2] Thus, there is considerable interest in developing *in situ* (in place) methods of separating out the bitumen. A wide range of possible in situ techniques have been suggested.[3]

One of the proposed in situ techniques is steam flooding. In the steam flood method an array of injection and production wells are drilled, then steam is pumped down the injection wells and oil and water are pumped up the production wells. A major problem with this method is in developing "communication" between the injection and the production wells. The unheated bitumen is so viscous it blocks the steam and heated, mobilized bitumen from traveling from one well to the other. Various methods have been proposed to develop communication. These include injection of solvents, formation of emulsions and formation fracturing.[17] The first two methods are designed to reduce the viscosity of the bitumen, and the third method produces a direct channel between wells.

The electrical preheat in situ method proposes that the oil sand formation be heated by passing an electrical current through it, and, once the vicosity has been lowered sufficently, steam or some other hot fluid be used to drive out the bitumen. The viscosity of bitumen decreases rapidly with increases in temperature.[4] At natural formation temperatures Athabasca bitumen has a viscosity in the millions of centipoise. If this bitumen is heated to 80°C the viscosity drops to 1000 cp, and if heated to 200°C the viscosity will be about 10 cp. The heat used to raise the temperature of the bitumen is generated right in the oil sand by the electrical current flowing in the connate water. This heating in place overcomes some of the problems usually associated with heating heavy oil formations with steam, such as low thermal conductivity and difficulty directing the applied heat. Electrical heating has seen limited use in secondary recovery of conventional oil fields.[5] Several patents have been issued on the application of electrical heating to oil sand. Petro-Canada Explorations Ltd. is running a pilot plant investigating the electrical preheat method.[4]

1.2 Means of Investigation

Before pilot plant tests or commercial developments of the electrical preheat method are embarked upon it is necessary to develop an idea of the optimal electrode configuration, heating rates, and heating times. Other than pilot plant or field tests, the study of the electric preheat method may be carried out by three means: analytic calculations, laboratory scale models, and numerical simulation. In all three of these means it is necessary that a mathematical model of the important processes be developed.

The electrical conductivity of oil sand is temperature dependent. This couples the governing thermal and electrical equations. The resulting set of partial differential equations is difficult or impossible to solve analytically for all but the simplest one dimensional cases.

Physical scale modeling of in situ recovery methods is discussed by Farouq Ali and Redford,[6] and the physical modeling of electromagnetic heating of oil sand is discussed by Vermeulen, Chute, and Cervenan.[7]

It was decided to use numerical simulation to study some aspects of the electrical preheat method. Two other studies reporting numerical simulation of electrical heating of petroleum reservoirs are by Todd and Howell[8] and El-Feky.[9] Todd and Howell simulate electrically heating oil sand to several hundred degrees Celsius, while cooling the well bore. El-Feky studied electrically heating conventional oil reservoirs for secondary recovery. Both of these studies and simulators differ in detail and application from this study.

Numerical simulation may be used as a compliment or alternative to physical scale modeling. Numerical simulation allows easy adjustment of parameters and electrode configurations, but is only as good as the mathematical model on which the simulation is based. Comparison of numerical simulation results and physical scale model results can provide insight into the accuracy of a mathematical model. Physical scale modeling of the electrical heating of oil sand is being done by Vermeulen, Chute and Cervenan[7] at the University of Alberta, and some of the results of their group are compared to the results obtained by the numerical simulator which is discussed in this thesis.

1.3 Electrical and Thermal Properties of Oil Sand

The electrical conductivity and relative dielectric constant of reconstituted oil sand were measured by Chute, Vermeulen, Cervenan and McVea[10] and correlated to frequency, moisture content, temperature, and density. Similarly, the heat capacity and thermal conductivity of various grades of oil sand were reported by Cervenan, Vermeulen, and Chute[11]. The applicable results of these two papers are summarized in Table 1.1 and Table 1.2.

Table 1.1 Electrical Properties of Reconstituted Oil Sand at 60 Hz and 24°C.[10]

Material	Water (% wt.)	Conductivity (S/m)	Dielectric Constant
Oil Sand A	5.8	1.5 X 10^{-2}	4. X 10^{4}
Oil Sand B	3.3	6. X 10^{-3}	1. X 10^{4}
Oil Sand C	1.3	1.2 X 10^{-3}	1. X 10^{3}

Table 1.2 Thermal Properties of Reconstituted Oil Sand and Other Materials.[11] [18]

Material	Water (% wt.)	Conductivity (W/m-K)	Heat Capacity (J/m^3-K)
Oil Sand	4.3	1.80	-
Oil Sand	2.1	1.42	-
Oil Sand	1.3	1.35	-
Oil Sand	1.5	-	1.71 X 10^{6}
Oil Sand	11.1	-	2.23 X 10^{6}
Oil Sand	1.4	-	1.83 X 10^{6}
Sandstone	-	.877	1.59 X 10^{6}
Limestone	-	1.70	1.86 X 10^{6}
Shale	-	1.04	1.87 X 10^{6}

The electrical conductivity was found to increase approximately linearly with temperature dependence up to 90°C. The computer program written in this study incorporates this temperature dependence. The electrical conductivity is expressed by

$$\sigma(T) = \sigma_{24}[1 + \alpha(T - 24)] \qquad (1.1)$$

where T is in degrees Celsius and σ is in Siemens/meter. The constant α has units $^\circ C^{-1}$ and is the temperature dependence of electrical conductivity. A typical measured value of α for oil sand is 2.3×10^{-2} $^\circ C^{-1}$.

The author knows of no published data on the temperature dependence of conductivity of oil sand at temperatures above 90°C. Chute et.al.[10] report that above 90°C the conductivity tends to level off, and hence the value of α decreases. In an electrical preheat of an oil sand formation the temperature will remain below 90°C in the bulk of the formation. Near the electrodes the temperature may reach 250°C or higher, but the conductivity of the oil sand will probably be affected by the injection of brine into the formation. Due to the lack of published data the author has assumed that the conductivity increases linearly with temperature, even well above 90°C.

While the thermal conductivity and heat capacity of oil sand may have some temperature dependence, there is no detailed study on such temperature dependence reported in the literature. A few measurements by Cervenan et. al.[11]

show no marked temperature dependence for the heat capacity of oil sand samples between 20°C and 70°C. The computer program was written so that the temperature dependences of thermal conductivity and heat capacity could easily be included once these dependences are known.

The effects on the electrical preheat process of changing the ratio of the electrical conductivities of the oil sand and surrounding formations is one of the items studied in this thesis. For the study of the effects of changing this ratio the thermal properties of the surrounding formations are assumed to be the same as those of the oil sand. It is also possible to give the surrounding formations different thermal properties than the oil sand.

1.4 The Computer Program "MEGAERA"

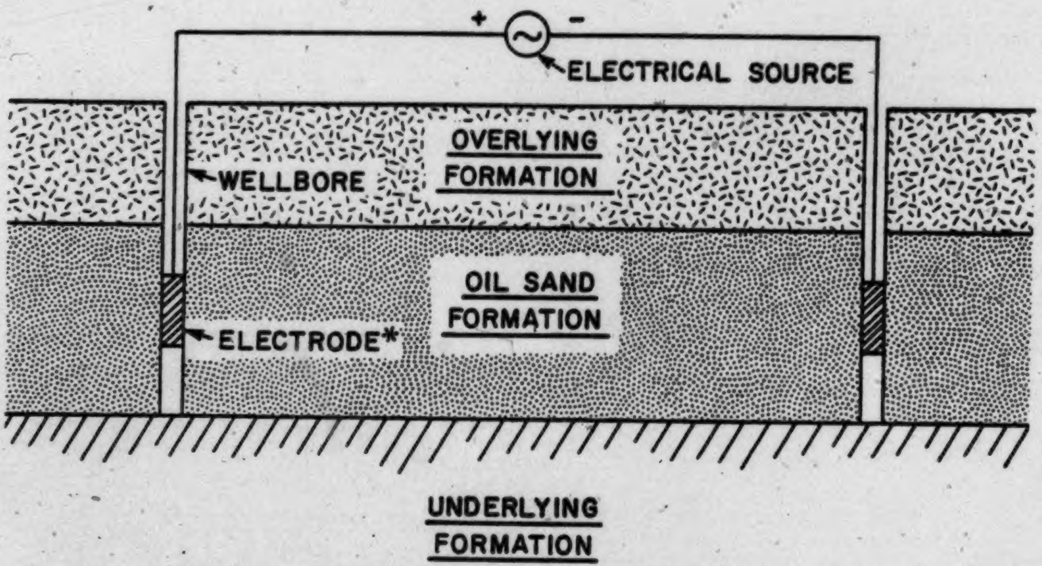
A computer program entitled MEGAERA[16] was written to numerically simulate the electrical conduction heating of oil sands, or other solid materials. The mathematical model and finite difference scheme used in MEGAERA are discussed in detail in the second and third chapters of this thesis. MEGAERA was used to study a variety of electrical heating configurations with different conductivity ratios between the oil sand and the surrounding formations. The results of this study are given in chapter four of this thesis.

Any study, analytic, experimental or numerical, must be limited to some extent. The specification for MEGAERA were based on the type of physical scale modeling being done by

Vermeulen, Chute and Cervenan at the University of Alberta and the experience of the author in writing and using an earlier two dimensional electrical heating simulator.

There were two primary considerations in developing MEGAERA. First, the program had to be able to simulate overburden and underburden of different electrical conductivities than the oil sand. The optimal electrode configuration for heating an oil sand formation is highly dependent on whether the surrounding formations are more or less conductive than the oil sand. Second, the program must be relativly inexpensive to run. For this reason MEGAERA was limited to two dimensions. A typical MEGAERA run takes two to three minutes of C.P.U. time on an Amdahl 470V/7. The time required by a three dimensional program can be conservativly estimated by multiplying the time required by the two dimensional program by the number of grid lines in the third dimension. For a reasonable amount of detail in the third dimension the program would be too expensive to run more than a few times.

While the universe is generally perceived to have three spacial dimensions, it is often convenient and reasonably accurate to model a physical system as two dimensional. In studying electrical heating of oil sand there are two different two dimensional slices of interest. A horizontal (areal) study of the heating pattern in a five spot pattern assumes that the oil sand formation is infinitly thick, but may yield results which are correct for the middle of a


+
-
ELECTRICAL SOURCE
WELLBORE
OVERLYING FORMATION
OIL SAND FORMATION
ELECTRODE*
UNDERLYING FORMATION

*For clarity the electrodes shown are of the simple cylindrical variety. In actual practise extended electrodes would be used.

Figure 1.1 Typical Vertical Electrode Geometry

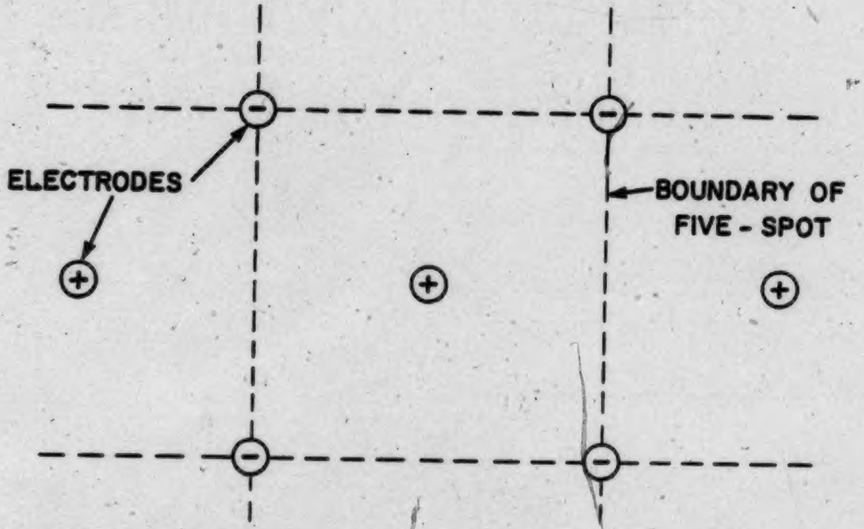

ELECTRODES
BOUNDARY OF FIVE - SPOT
+
-

Figure 1.2 Typical Horizontal Electrode Geometry

finite formation. A vertical section studying the heating between two parallel plates allows one to see the effects of changing the conductivity of the surrounding formations. Parallel plates may be approximated in a field test by systems of closely spaced electrodes or by using a more widely spaced system of electrodes in which each electrode has been effectively enlarged by jet cutting the formation with a brine solution.

In modeling configurations with overburden and underburden the energy losses due to electrical current leakage and heat conduction from the oil bearing formation are calculated by MEGAERA. Voltage, current, power and resistance calculation are done at each time step. MEGAERA will automatically scale the applied voltages to produce constant power and constant current heating, if this is desired. A post processor, called MEGAERA.PLOTTER, was written to produce contour and profile plots of the electrical potential, heating rate and temperature at specified times.

2. The Mathematical Model

2.1 Electrical Fields

The basic set of equations of classical electricity and magnetism, first written by Maxwell in 1863, are:[12]

$$\nabla \cdot \bar{D} = \rho \tag{2.1}$$

$$\nabla \cdot \bar{B} = 0 \tag{2.2}$$

$$\nabla \times \bar{E} = -\frac{\partial \bar{B}}{\partial t} \tag{2.3}$$

$$\nabla \times \bar{H} = \bar{J} + \frac{\partial \bar{D}}{\partial t} \tag{2.4}$$

These equations, together with the appropriate boundary conditions, may be used to find the electric field of any electrical heating configuration at any frequency. However, an analytic or numerical solution to Maxwell's equations is difficult, if not impossible, in all but the simplest cases. It is therefore desirable to simplify these equations, dropping terms which are negligible for the frequencies and electrode configurations of interest.

2.1.1 The Quasi-static Approximation

In their paper on physical scale modeling, Vermeulen et.al.[7] discuss the simplification of Maxwell's equations for various types of electrical heating. The discussion in this section (2.1.1) is drawn from the work of their paper and from Magid.[13]

Consider the displacement current term in equation (2.4). The ratio of conduction current to displacement current is given by $\sigma/\omega\varepsilon$ for time harmonic solutions. For oil sand this ratio is typically greater then one hundred at power frequencies (60 Hz), and does not fall to ten until frequencies of about one megahertz are reached. Thus, for power frequencies the effects of displacement current may be neglected.

The study of the fields of two dimensional, parallel electrode, electrical heating configurations is similar to the calculation of electric fields in lossy parallel plate capacitors. In these types of problems the effects of the time varying magnetic field on the time varying electric field may be neglected if the largest dimension of the problem geometry is much less than a wavelength in the medium. This reduction of Maxwell's equations to the static field equations by neglecting these magnetic and displacement current effects is known as the quasi-static approximation.

The wavelength in oil sand at 60 Hz is between 2000 m and 9000 m,[7] depending on the water content of the oil sand. As the maximum dimension of a typical electrical heating configuration will probably be less then 200 m, the quasi-static approximation should be valid. If the surrounding formations (overburden and underburden) are also to be modeled, the wavelength in these materials must also be much greater then the maximum problem dimension and the loss tangent must be much greater than one.

When using the quasi-static approximation the geometry of the electric field is calculated from the static electric field equations. While calculating the fields the potential difference between electrodes is fixed at the peak A.C. value. The time varying electric field is then expressed as:

$$\bar{E} = \bar{E}(x,y,z) \sin \omega t \qquad (2.5)$$

where $\bar{E}(x,y,z)$ describes the static field, $\omega = 2\pi f$ where f is the frequency, and t is the time in seconds. The r.m.s. value of the potential difference is used in MEGAERA rather then the peak value. This simplifies the calculation of heating rates and conforms to the practise of stating power line voltages as r.m.s. rather then peak values.

The discussion in the previous paragraph applies to single phase heating. When three phase heating is used the geometry of the electric field of each phase must be calculated separately, using the static field equations and setting the electrodes of the other phases to ground potential. Each of the three electric field geometries thus calculated is then multiplied by the appropriate time variation, which must include the phase differences. Finally, the three fields are added vectorially and the square of the magnitude of the resulting field is averaged over a cycle to get the heating rate. This procedure was not incorporated in MEGAERA, so multiphase heating cannot be simulated.

2.1.2 Current Continuity Equation

Neglecting the effect of the time varying magnetic field of the electric field yields:

$$\nabla \times \bar{E} = 0 \tag{2.6}$$

Thus, a scalar potential $\psi(x,y,z)$ may be defined.[14]

$$\bar{E} = -\mathrm{grad}\,\psi \tag{2.7}$$

Droping the displacement current term in equation (2.4) gives

$$\nabla \times \bar{H} = \bar{J} \tag{2.8}$$

Taking the divergence of both sides of equation (2.8) gives

$$\mathrm{div}\,\bar{J} = 0 \tag{2.9}$$

which states that the current is continuous. As $\bar{J} = \sigma\bar{E}$, where σ is the electrical conductivity, equation (2.9) becomes

$$\mathrm{div}(\sigma\,\mathrm{grad}\,\psi) = 0 \tag{2.10}$$

As the electrical conductivity σ depends on temperature, it will, in general, be a function of of position, and equation (2.10) may not be simplified by moving the conductivity term outside the divergence operator.

2.1.3 Electrical Boundary Conditions

Three different types of electrical boundaries are to be modeled in the electrical conduction heating simulation. They are: electrode - conducting material boundaries, insulator - conducting material boundaries, and boundaries between two conducting materials.

Electrodes are simply modeled as regions of constant voltage. If constant power or constant current heating is requested, MEGAERA first calculates the fields assuming constant voltage, then measures the current and scales the electric field to the appropriate case. Electrode - Oil Sand contact impedances were noted by Chute et. al.[10] when measuring the electrical properties of oil sand. A major cause of contact impedence is poor physical contact between the oil sand and the electrode. In the mathematical model used it is assumed that contact impedance has been made negligible by a suitable technique such as the injection of a saline solution near the electrode.

Insulator - conducting material boundaries are modeled in electrostatics as surfaces at which there is no current flow normal to the surface (a homogeneous Neumann boundary condition). Having no current normal to the surface implies that the normal component of the electric field is zero just inside the conducting material. As well as modeling insulating boundaries, this type of boundary condition may be used to model planes of symmetry. Many of the electrode configurations of interest in electrical conduction heating of oil sand consist of large arrays of electrodes of alternating polarity. The planes of symmetry in the electric field may be used to divide the large array into many small identical cells. The simulation of the heating in one of these small cells is all that is necessary to find the heating in the entire array.

At a boundary between two media of finite conductivity the electric field boundary conditions are:[15]

$$\bar{n} \times (\bar{E}_2 - \bar{E}_1) = 0 \qquad (2.11)$$

$$\bar{n} \cdot (\bar{D}_2 - \bar{D}_1) = \rho_s \qquad (2.12)$$

where $\bar{E}_1$, $\bar{E}_2$, and $\bar{D}_1$, $\bar{D}_2$, are the fields in the two media, and where ρ_s is the surface charge at the interface in coulombs per square meter and $\bar{n}$ is a unit vector locally normal to the boundary. These are general boundary conditions for Maxwell's equations and are valid for all frequencies. The boundary conditions may be expressed in terms of the normal and tangential components of the electric field.

Starting with Maxwell's equations, take the divergence of equation (2.4) and substitiute in equation (2.1) to get the current continuity equation.

$$\nabla \cdot \bar{J} + \frac{\partial \rho}{\partial t} = 0 \qquad (2.13)$$

Writing this equation in its integral form and applying the divergence theorem gives

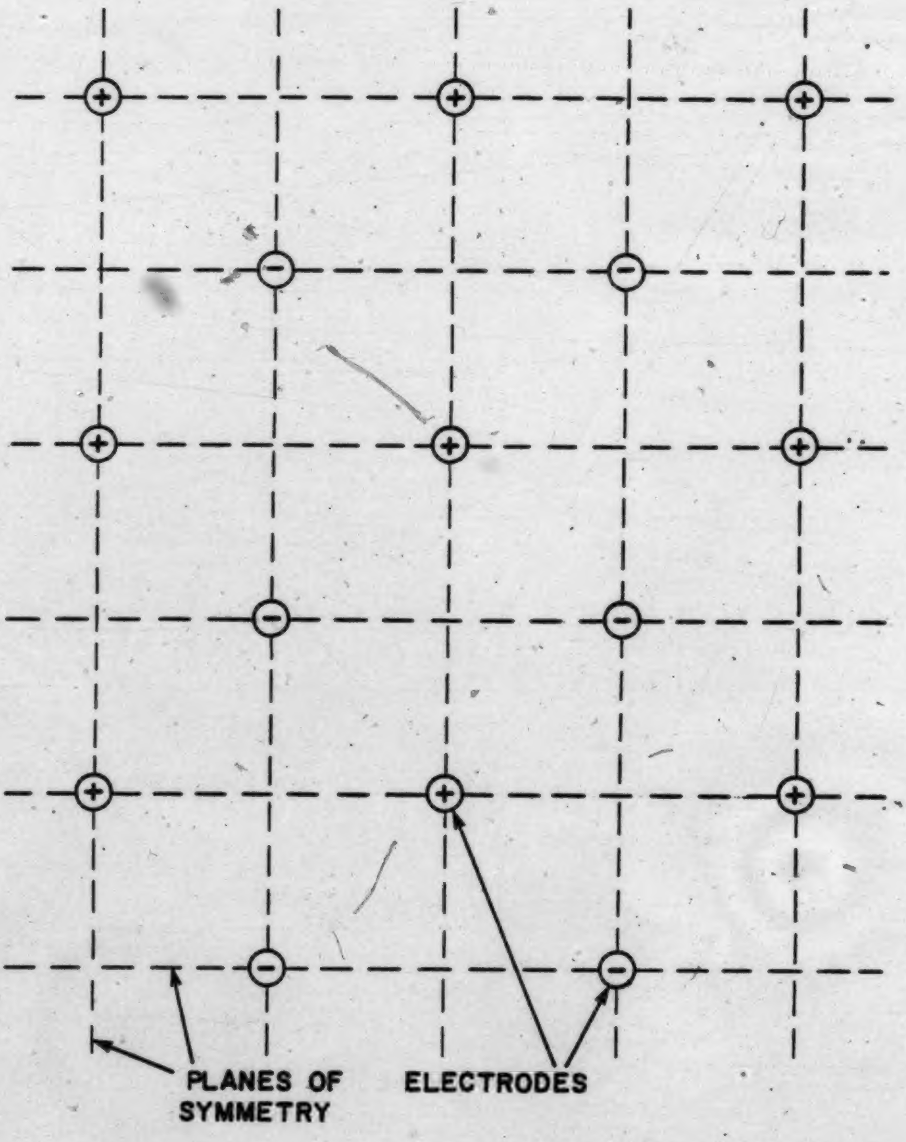

PLANES OF SYMMETRY
ELECTRODES
+
−

Figure 2.1 Planes of Symmetry in a Five Spot Array

$$\int_S \bar{J} \cdot \bar{ds} = -\int_V \frac{\partial \rho}{\partial t} \, dv \qquad (2.14)$$

To apply equation (2.14) to the boundary between two materials of finite conductivity, consider a small cylindrically shaped volume enclosing a portion of the boundary. One face of the cylinder, denoted S_1, is in material 1 and is parallel to the boundary. The second face, S_2, is in material 2 and is parallel to the boundary. The side of the cylinder, S_3, cuts across the boundary connecting the two faces. Applying equation (2.14) now gives

$$\int_{S_1} \bar{J} \cdot \bar{ds} + \int_{S_2} \bar{J} \cdot \bar{ds} + \int_{S_3} \bar{J} \cdot \bar{ds} = -\int_V \frac{\partial \rho}{\partial t} \, dV \qquad (2.15)$$

Now, let the length of the side of the cylinder shrink toward zero. As both materials 1 and 2 are of finite electrical conductivity, the current density must remain finite, and the integral over S_3 goes to zero. For S_1 and S_2 sufficently small (but an order larger then S_3), equation (2.15) becomes

$$\sigma_1 E_{n1} \Delta S + \sigma_2 E_{n2} \Delta S = -\frac{\partial \rho_s}{\partial t} \Delta S \qquad (2.16)$$

where Δ S is the area of a face of the cylinder, ρ_s is the surface charge density per unit area, and E_{n1} and E_{n2} are the components of the electric field normal to the boundary in materials 1 and 2 respectively. Dividing out the ΔS and assuming a time harmonic field yields

$$\sigma_1 E_{n1} - \sigma_2 E_{n2} = -j\omega\rho_s \qquad (2.17)$$

From equation (2.12) and assuming that the electrical permittivity is uniform and isotropic in each material

$$\varepsilon_1 E_{n1} - \varepsilon_2 E_{n2} = \rho_s \qquad (2.18)$$

Substituting in equation (2.17) gives

$$(\sigma_1 + j\omega\varepsilon_1) E_{n1} = (\sigma_2 + j\omega\varepsilon_2) E_{n2} \qquad (2.19)$$

Recall that for the quasistatic approximation to hold $\sigma/\omega\varepsilon$ must be much greater then one. Thus, when using the quasistatic approximation the normal component of current is continuous across a boundary between two materials of finite electrical conductivity.

Treating the problem as quasistatic and assuming there are no double layers of charge at the interface, the electric potential must be continuous across the boundary. Thus the boundary conditions used are

$$\sigma_1 E_{n1} = \sigma_2 E_{n2} \qquad (2.20)$$

$$\psi_1 = \psi_2 \qquad (2.21)$$

In this discussion of the boundary between two materials of finite conductivity it was assumed that the conductivity was discontinuous at the boundary. In an actual oil sand formation the conductivity may change continuously between the oil sand and the overburden or underburden. The electric fields calculated assuming either a sharp, discontinuous boundary, or assuming a continuous transition over a short distance will be the same except in the transition region. As no data is available describing how the conductivity changes from the oil sand to the surrounding formations, and as the transition distance is likely small compared to the formation thickness, it is assumed that a sharp, discontinuous change in conductivity occurs at the boundary.

2.2 Heat Generation and Tranfer

The instantaneous heating rate due to electrical conduction is given by

$$\dot{Q} = \sigma \bar{E} \cdot \bar{E} \tag{2.22}$$

If, when using the quasi-static approximation, the r.m.s. values of the applied voltages are used, then the average heating rate is

$$\dot{Q}_{AV} = \sigma |\bar{E}_{rms}|^2 \tag{2.23}$$

Of the three major heat transfer mechanisms (radiation, convection, and conduction) only conduction is included in the mathematical model. Radiation is negligible as the radiation from a black body at reservoir temperatures would only penetrate a very short distance into the oil sand. Natural convection due to temperature gradiants is neglected even though its importance as a heat transfer mechanism in oil sand at the temperatures of interest is unclear. The scale modeling factors developed by Vermeulen et. al.[7]

included only conduction in their derivation. By comparing the results of numerical simulations based on this premise with the results of physical scale model runs some grasp on the accuracy of this mathematical model may be obtained. The treatment of forced convection due to fluid injection or pumping is beyond the scope of this study.

Thus, the equation used to model the heat generation and transfer is

$$M\frac{\partial T}{\partial t} = \nabla\cdot(k_h \nabla T) + \sigma|\bar{E}_{rms}|^2 \qquad (2.24)$$

where M is the volumetric heat capacity, K_h is the thermal conductivity, and T is the temperature in Celsius. Note that in the computer program the volumetric heat capacity and thermal conductivity are assumed constant within each formation.

The outer boundaries are assumed to be insulated, no-heat-flow boundaries. If the boundaries are planes of symmetry or actual insulated boundaries this is a good model. However, if the problem to be modeled consists of an oil sand formation surrounded above and below by formations which are much thicker then the oil sand formation (i.e. semi-infinite overburden and underburden) artificial boundaries will have to be introduced to limit the extent of the problem domain. If these boundaries are far enough from the electrodes little current or heat will reach the area near these artificial boundaries.

Different materials in the problem domain may have different thermal properties. At the boundary between two different materials the temperature and heat flow across the boundary are assumed to be continuous.

3. Numerical Solution of the Model

Finite difference techniques are used to approximate the solution of partial differencial equations in many different fields, including oil reservoir simulation, laser-plasma interaction studies, and water flow problems. Finite difference approximations to derivatives were used by L. Euler (1768) and there are currently several textbooks available which explain the finite difference method in detail.[19] [20] [21] The error, stability and convergence of the approximation and solution techniques are areas of concern and study when using this method to solve partial differential equations. In this chapter the particular methods used in developing MEGAERA are outlined, and an explanation of how boundary conditions, stability and convergence problems were delt with is given.

3.1 Differencing of the Current Continuity Equation

The electrical current continuity equation (equation 2.10) may be rewritten (for two dimensions) as

$$\frac{\partial}{\partial x}\left(\sigma \frac{\partial \psi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\sigma \frac{\partial \psi}{\partial y}\right) = 0 \qquad (3.1)$$

The finite difference approximation for the first term of equation (3.1) is given by

$$\frac{\partial}{\partial x}\left(\sigma \frac{\partial \psi}{\partial x}\right) \cong$$
$$\left(\frac{2}{h_i + h_{i-1}}\right)\left[\left(\frac{\sigma_{i+1}+\sigma_i}{2}\right)\left(\frac{\psi_{i+1}-\psi_i}{h_i}\right) - \left(\frac{\sigma_{i-1}+\sigma_i}{2}\right)\left(\frac{\psi_i-\psi_{i-1}}{h_{i-1}}\right)\right] \qquad (3.2)$$

See figure 3.1 for details on the grid structuring and labeling.

This approximation (equation 3.2) is "conservative" in that the current flowing out of the left boundary of the grid block (i,j) is equal to the current flowing into the right boundary of the grid block (i+1,j). A symmetric approximation for the second (y direction) term of equation (3.1) is added to equation (3.2) to give an algebraic difference equation at the center of each grid block. Rearranging the terms gives

$$R_{ij}\psi_{i,j+1} + S_{ij}\psi_{i+1,j} + A_{ij}\psi_{ij} + C_{ij}\psi_{i-1,j} + F_{ij}\psi_{i,j-1} = 0 \qquad (3.3)$$

where

$$R_{ij} = (\sigma_{i,j+1} + \sigma_{ij})/[k_j(k_j + k_{j-1})] \qquad (3.4)$$

$$S_{ij} = (\sigma_{i+1,j} + \sigma_{ij})/[h_i(h_i + h_{i-1})] \qquad (3.5)$$

$$A_{ij} = -R_{ij} - S_{ij} - C_{ij} - F_{ij} \qquad (3.6)$$

$$C_{ij} = (\sigma_{i-1,j} + \sigma_{ij})/[h_{i-1}(h_i + h_{i-1})] \qquad (3.7)$$

$$F_{ij} = (\sigma_{i,j-1} + \sigma_{ij})/[k_{j-1}(k_j + k_{j-1})] \qquad (3.8)$$

The presence of a boundary adjacent to the grid block requires the modification of equations (3.3) to (3.8). If an insulator is in the (i-1,j) grid block, then the boundary condition in the left direction is $\frac{\partial\psi}{\partial n} = 0$. This is

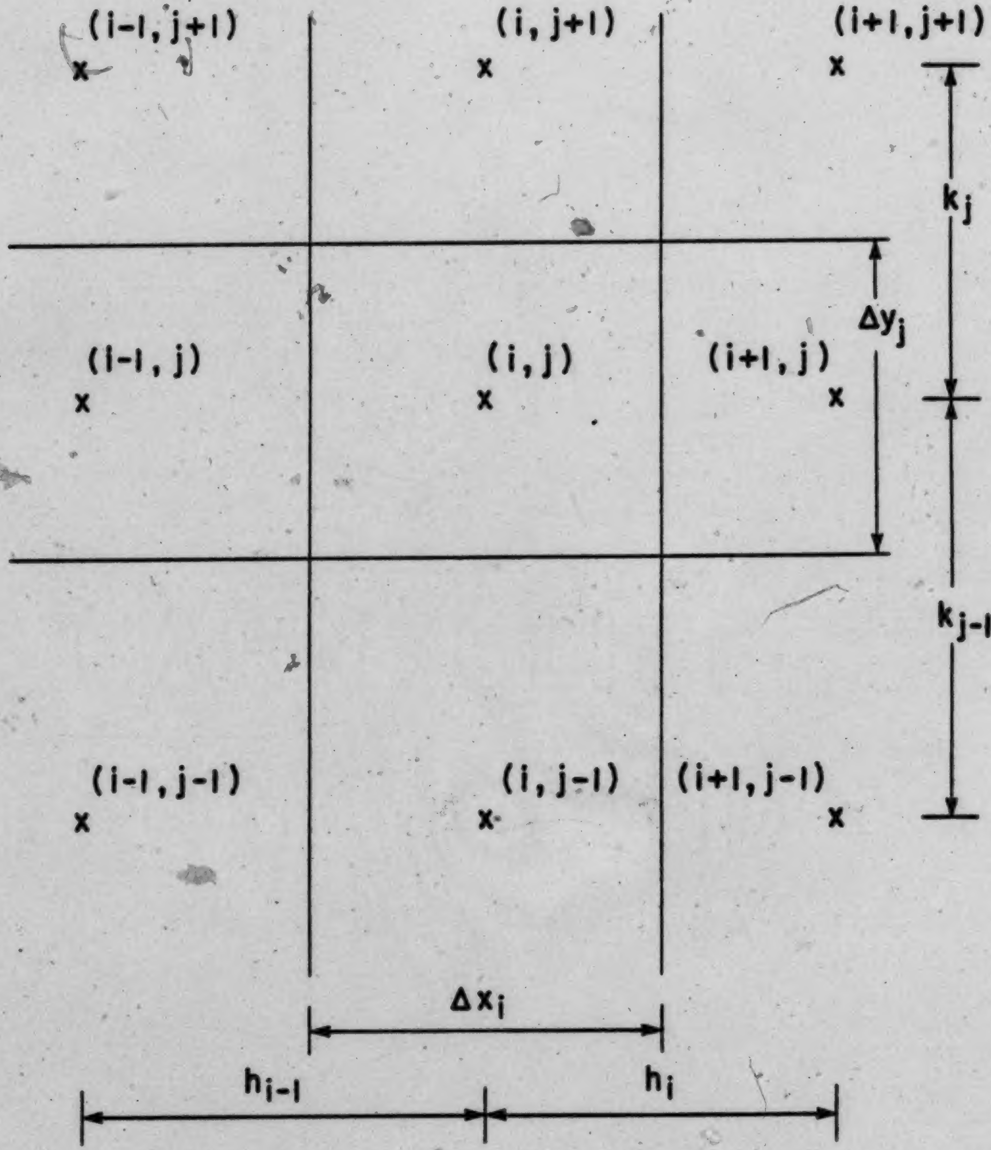

(i-1, j+1)
(i, j+1)
(i+1, j+1)
k_j
Δy_j
(i-1, j)
(i, j)
(i+1, j)
k_{j-1}
(i-1, j-1)
(i, j-1)
(i+1, j-1)
Δx_i
h_{i-1}
h_i

Figure 3.1 Finite difference grid used in MEGAERA showing the locations at which potential, temperature and conductivities are computed.

approximated by setting $\psi_{i-1,j}$ equal to $\psi_{i,j}$. Equation (3.3) becomes

$$R_{ij}\,\psi_{i,j+1} + S_{ij}\,\psi_{i+1,j} + (A_{ij} + C_{ij})\,\psi_{i,j} + F_{ij}\,\psi_{i,j-1} = 0 \qquad (3.9)$$

If the (i-1,j) grid block is part of an electrode then $\psi_{i-1,j}$ is known. The coefficent of $\psi_{i-1,j}$ must be modified to take into account the infinite conductivity of the electrode (which extends up to the grid block boundary). Thus, the difference equations are

$$C'_{ij} = \sigma_{ij}/[\Delta x_i (h_i + h_{i-1})] \qquad (3.10)$$

$$A'_{ij} = -R_{ij} - S_{ij} - C'_{ij} - F_{ij} \qquad (3.11)$$

$$R_{ij}\,\psi_{i,j+1} + S_{ij}\,\psi_{i+1,j} + A'_{ij}\,\psi_{ij} + F_{ij}\,\psi_{i,j-1} = -C'_{ij}\,\psi_{i-1,j} \qquad (3.12)$$

When a medium of different conductivity is in the (i-1,j) grid block the equations (3.3) to (3.8) must be modified to deal explicitly with the conductivity transition. Equations (2.20) and (2.21) give the boundary conditions at the interface. If the electric field is constant between the center of the grid block and the interface, the continuity condition for the electrical potential (Equ. 2.21) yields

$$(\psi_{ij} - \psi_{i-1,j}) = \frac{\partial \psi_{ij}}{\partial x}\left(\frac{\Delta x_i}{2}\right) + \frac{\partial \psi_{i-1,j}}{\partial x}\left(\frac{\Delta x_{i-1}}{2}\right) \qquad (3.13)$$

Figure (3.2) shows a profile of ψ vs X between the grid block centers. Writing equation (2.20) as

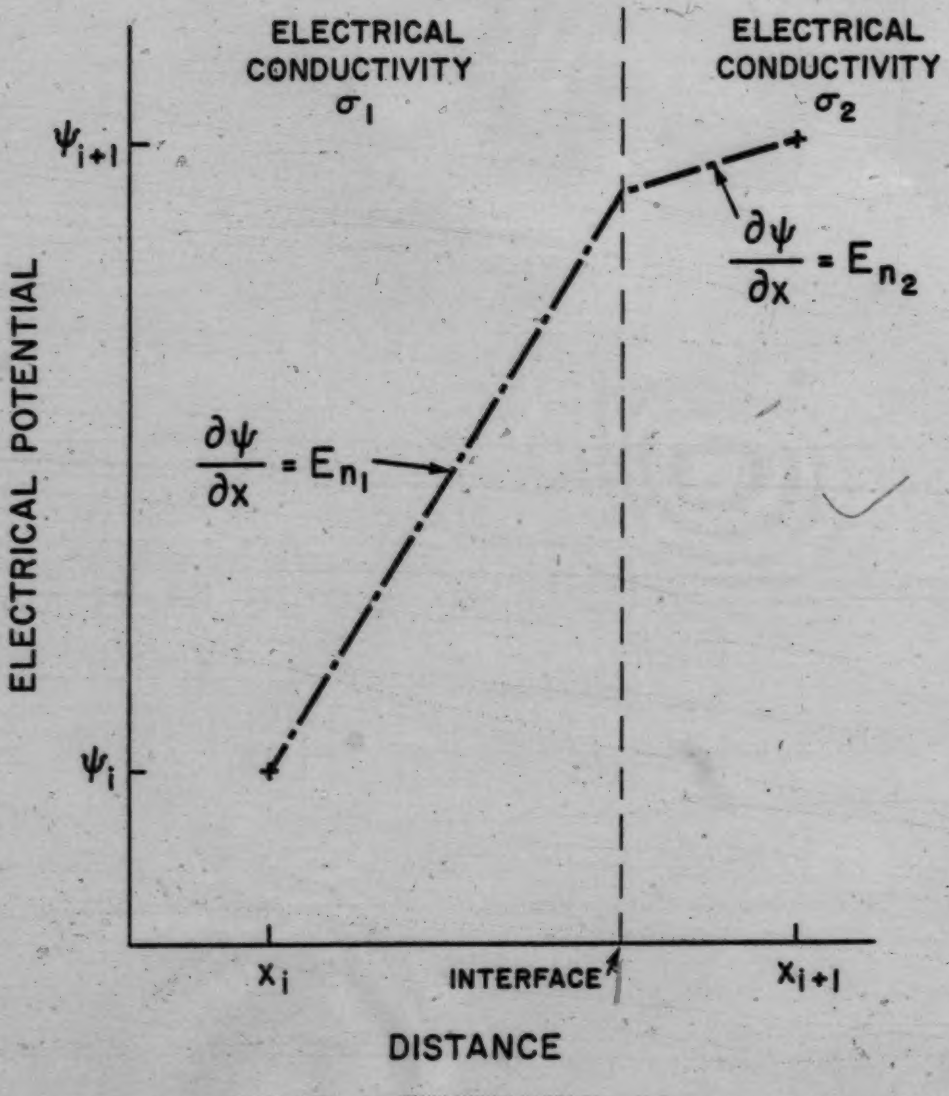

ELECTRICAL CONDUCTIVITY σ_1
ELECTRICAL CONDUCTIVITY σ_2
ψ_{i+1}
ψ_i
$\frac{\partial\psi}{\partial x} = E_{n_2}$
$\frac{\partial\psi}{\partial x} = E_{n_1}$
ELECTRICAL POTENTIAL
x_i
INTERFACE
x_{i+1}
DISTANCE

Figure 3.2 Profile of Electrical Potential ψ Across an Conductivity Discontinuity, showing the abrupt change in normal component of the electric field as required for continuity of the normal component of current

$$\sigma_{i,j}\frac{\partial\psi_{i,j}}{\partial x} = \sigma_{i-1,j}\frac{\partial\psi_{i-1,j}}{\partial x} \tag{3.14}$$

the derivative $\frac{\partial\psi_{i-1,j}}{\partial x}$ in (3.13) may be eliminated.

$$\frac{\partial\psi_{i,j}}{\partial x} = (\psi_{i,j}-\psi_{i-1,j})/\left[\frac{\Delta x_i}{2} + \left(\frac{\sigma_{i,j}}{\sigma_{i-1,j}}\right)\frac{\Delta x_{i-1}}{2}\right] \tag{3.15}$$

Thus, the coefficents C_{ij} and A_{ij} in the finite difference equation (3.3) become

$$C''_{ij} = 4\sigma_{ij}/\left[(h_i+h_{i-1})\left(\Delta x_i + \frac{\sigma_{i,j}}{\sigma_{i-1,j}}\Delta x_{i-1,j}\right)\right] \tag{3.16}$$

$$A''_{ij} = -R_{ij} - S_{ij} - C''_{ij} - F_{ij} \tag{3.17}$$

3.2 Differencing the Thermal Equation

The thermal partial differential equation (2.24) is in many respects analogous to the electrical current continuity equation (2.10). The differencing of the term

$$\frac{\partial}{\partial x}\left(k_h\frac{\partial T}{\partial x}\right) \cong$$

$$\frac{2}{(h_i+h_{i-1})}\left[\left(\frac{k_{h\,i+1,j}+k_{h\,i,j}}{2}\right)\left(\frac{T_{i+1,j}-T_{i,j}}{h_i}\right) - \left(\frac{k_{h\,i-1,j}+k_{h\,i,j}}{2}\right)\left(\frac{T_{i,j}-T_{i-1,j}}{h_{i-1}}\right)\right] \tag{3.18}$$

is directly analogous to the differencing of the x term in equation (2.10). The handling of the boundary conditions is also analogous, with constant temperature boundaries corresponding to electrodes and thermal insulators corresponding to electrical insulators.

The time derivative of the thermal equation is approximated by the forward difference

$$M \frac{\partial T}{\partial t} \cong M_{ij}(T_{ij}^{n+1} - T_{ij}^{n})/\Delta t^{n} \qquad (3.19)$$

where the superscript indicates the number of the timestep. The heating rate term is also calculated and included in the complete difference equation.

3.3 Solution of the Difference Equations

In solving a set of partial differential equations by the finite difference technique it is desirable to reduce the set of equations by eliminating all but one unknown (one unknown being a single variable unknown at all points in the domain). This may be done before or after differencing the equations. In the electrical heating simulation presented in this thesis, the electrical and thermal equations are coupled non-linearly making the elimination of either the potential or the temperature difficult. For this reason the equations were coupled explicitly: first the electric field was found, then the new temperatures were found. This explicit coupling leads to convergence problems which are discussed in the section on program testing.

The procedure followed by MEGAERA for each time step is as follows:

1. The electrical conductivity of each grid block is updated from the temperature of the grid block at the end of the last time step.
2. The coefficents of the electrical difference equation are found for each grid block.

3. This set of electrical difference equations is simultaneous. With one equation per grid block, a matrix of 2500 rows and columns is formed by MEGAERA if the full fifty by fifty grid is used. This matrix is solved by the Alternating Direction Implicit Procedure. This is an iterative relaxation method, which requires an initial guess at the solution. As the potential does not vary greatly from time step to time step, the potential found in the last time step is used as the initial guess for the potential of this time step and the new potential is found after only a few iterations.
4. With the potential now known, the volumetric heating rate in each grid block is found. MEGAERA calculates the heating rate directly from the potential and cannot output the electric field. The total current passing through the formation between the electrodes is calculated. If constant current or constant power heating is requested, the heating rates are scaled appropriately.
5. The temperatures at the end of the time step are found by solving the thermal equation using the Alternating Direction Implicit method. Despite the similarity of the names, the ADI method is different then the Alternating Direction Implicit Procedure, the former being used for parabolic equations and the latter for elliptic equations.
6. An energy balance is done for the time step. The energy

input, both for the time step and cumulative, is calculated in three different ways, as will be discussed in section 3.4.1. The energy calculations are done for each region of the domain separately, allowing one to find the percentage of energy input to, for example, the overburden and comparing it to the energy input to the oil sand.

MEGAERA was written to allow variable grid widths. The ratio of maximum width to minimum width should not excede ten in the domain of the problem, with adjacent grid widths being within a factor of two. The program is only two dimensional and the maximum number of grid blocks in each the X and Y directions is fifty. MEGAERA was written in single precision, which on the Amdahl 470 is 32 bit, hexidecimal format floating point arithmetic. Using the full fifty by fifty grid, MEGAERA uses approximately 1.9 seconds of C.P.U. time per time step on an Amdahl 470V/7, and requires about 110 pages of virtual memory. Thus, a typical eighty time step run will require about two and a half minutes of C.P.U. time.

3.4 Testing of MEGAERA

A large simulation code may be tested in several ways. Internal checks, such as energy balance calculations, give one indication of the accuracy of the program. Analytic solution of simple problems may be possible, and the results of simulating these problems may be compared to the analytic solution. Physical modeling of more complex problems may be possible and numerical simulation results may be compared to physical model results. Finally, one may check that alteration of the problem grid and scale do not significantly affect the results produced by the simulator. All of these methods were used to test MEGAERA.

3.4.1 Energy Balance

The cumulative electrical energy input at the electrodes is found from

$$E_E^n = E_E^{n-1} + V^n I^n \Delta t^n \tag{3.20}$$

where E_E^n denotes the total electrical energy input up till the end of the nth time step, V^n and I^n are the voltage and current respectively during the nth time step, and Δt^n indicates the length in seconds of the nth time step.

The cumulative electrical energy that is converted to heat is found by integrating the volumetric heating rate over the volume and time:

$$E_H^n = E_H^{n-1} + \sum_{ij} \dot{Q}^n_{AV_{i,j}} V_{i,j} \Delta t^n \tag{3.21}$$

where E_H^n is the cumulative energy converted to heat at the end of the nth time step, $V_{i,j}$ is the volume of (i,j) grid block and $Q^n_{AV_{i,j}}$ is the volumetric heating rate in the grid block during the nth time step.

The total heat energy stored in the temperature rise of the formation is found by summing the product of temperature rise, volumetric heat capacity and volume of each grid block:

$$E_T^n = \sum_{ij} (T_{ij}^n - T_o) M_{ij} v_{ij} \qquad (3.22)$$

All three calculations of the energy input to the formation should yield the same quantitative result. The computer program calculated E_E, E_H, and E_T for each timestep that was output. Comparing E_E and E_H, the differences between both the cumulative and the incremental values of these quantities were under one percent for all production runs of MEGAERA. The differences were greatest for the first time step, usually about .5 to .7 percent, and decrease as the run progressed. By the eightyth time step both cumulative and incremental differences were below one quarter percent. The difference between E_H and E_T was usually less than 0.1 percent. These figures are taken from production runs (Run No.s 10 - 20), where the electrical conductivity varied from one region to the next by up to a factor of ten.

3.4.2 Comparison with Analytic Results

A number of MEGAERA runs were done for simple one dimensional problems whose solutions could be found analytically. Initially the temperature dependence of electrical conductivity was set equal to zero, uncoupling the electrical equations from the thermal equation. The problem geometry consisted of two materials of different conductivity in series. The results showed that the potential solver, constant current or power scaling, heating rate calculator, and temperature solver were working properly, at least for one dimensional problems. In these tests the calculated values agreed with the analytic solutions to within .1 percent. The problems were solved using grids of varying grid width and differing overall scale (i.e. maximum dimensions).

It is also possible to analytically solve a one dimensional problem involving electrical conductivity which varies with temperature. If a rectangle of homogeneous material is placed between two parallel plates (and the remaining four sides are insulated) and a constant voltage is applied between the plates, the electric field will remain constant between the plates. As the conductivity rises with temperature so will the heating rate. The temperature at any time may be found by solving the integral equation

$$T = \frac{V^2 \sigma_{24}}{M d^2} \int_0^t [1 + \alpha (T - 24)]\, dt + T_0 \qquad (3.23)$$

where d is the distance between the plates, and V is the applied voltage (r.m.s. value). This integral equation may be solved by use of a Laplace transform. The solution is[22 23]

$$T(t) = \left[\frac{1}{\alpha} + (T_0 - 24)\right] \exp\left(\frac{V^2 \sigma_{24} \alpha}{M\ d^2}\, t\right) + 24 - \frac{1}{\alpha} \qquad (3.24)$$

Runs was carried out modeling this problem. When the maximum allowed temperature change per timestep was three percent, the difference between the computer solution and the analytic solution was one half of one percent after a temperature rise of 230°C. If the maximum allowed temperature change per timestep was increased to ten percent, this difference, after a similar temperature rise, was five percent. The former case required roughly three times the C.P.U. usage of the latter case.

The difference between the analytic solution and the computed temperatures is due to the explicit coupling of the electrical and thermal equations. The electrical field and heating rates are held fixed in a time step, resulting in a conservative approximation to the actual heating. As the conductivity increases with temperature, the calculated heating rate will always be less then its actual value. The error induced by this problem is negligible if the maximum allowed temperature change per timestep is five percent or less.

3.4.3 Comparison with Physical Models

MEGAERA was used to simulate a physical model run which was carried out by J. Fearn and A. Vogan of the Applied Electromagnetics group at the University of Alberta. The geometry of the physical model was a scaled down version of figure 4.2. The model consisted of eighteen inches (457 mm) of underlying sand covered by seven and three quarters inches (197 mm) of low conductivity oil sand, which in turn is covered by six inches (152 mm) of overburden sand. The spacing between the electrodes was thirteen inches (330 mm) and the third dimension of the model, i.e. the thickness of the two-dimensional slice, was four inches (102 mm). The electrodes were constructed of one quarter inch (6 mm) copper plate and were two inches (51 mm) high and their lengthes were equal to the thickness of the two-dimensional slice, i.e. four inches (102 mm). The electrodes were positioned so the right electrode was forty percent in the overburden sand and the left electrode was forty percent in the underlying sand, the remainder of each electrode being in the oil sand.

The underlying and overburden formations in the scale model were modeled using packed sand whose moisture content had been adjusted so the conductivity of the packed sand (density of 1.8 gm/cm^3) was 7.4 X 10^{-3} S/m. The oil sand was modeled with oil sand whose conductivity when packed (density of 2.0 gm/cm^3) was 2.0 X 10^{-3} S/m.

The scale model was heated by passing a constant current of .183 A between the electrodes for 3605 seconds. The initial temperature of the model was 22°C (±.8°). After the run was completed the thermal properties of the sand and oil sand used in the model were measured. The thermal conductivity of the underlying and overlying sand was 1.56 W/K-m and its volumetric heat capacity was 1.37 X 10^6 $J/K\text{-}m^3$. The thermal conductivity of the oil sand was 1.61 W/K-m and its volumetric heat capacity was 1.8 X 10^6 $J/K\text{-}m^3$.

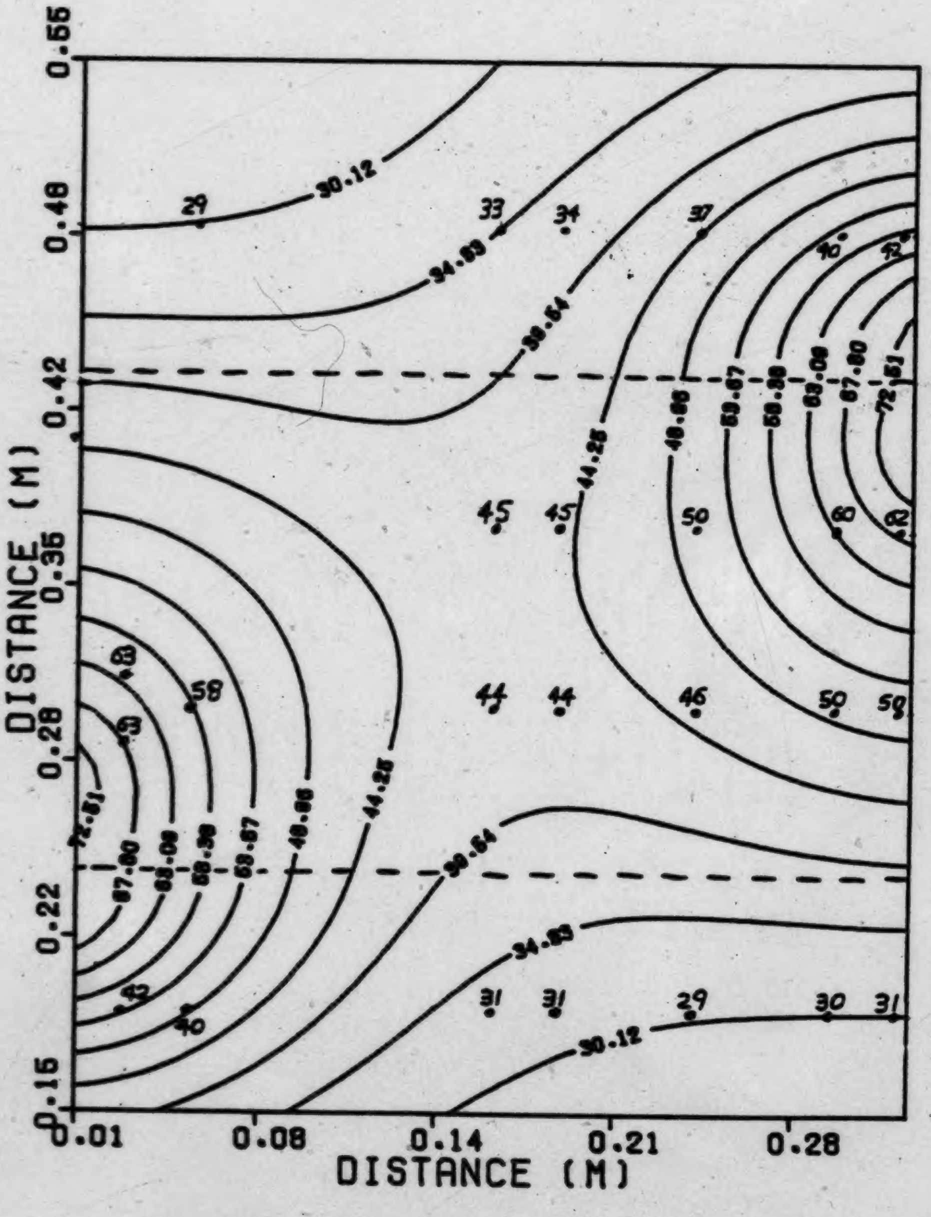


Figure 3.3 Computer calculated and measured temperatures for the physical model run. The contour plots are computer generated and the measured values are written on top.

The simulation was carried out with the thermal conducitity of the electrodes initially set to that of oil sand, and secondly to a value ten times greater. This change caused only a slight change in the final temperature profile in the immediate vicinity of the electrodes.

The geometry, dimensions, sand and oil sand electrical and thermal properties, and the heating type and time were input to MEGAERA. The temperature dependence of electrical conductivity was not measured, but was assumed to be 0.0229 K^{-1}, which is a typical value at low frequencies[7]. A contour plot of the simulation results is given in figure 3.4 with the measured temperatures from the physical model noted where measured. Agreement between the simulation results and the physical model results is excellent, except in the immediate vicinity of the electrodes. The measured and computed resistances were also compared and the percent difference between the two was always less then two percent.

4. Results of MEGAERA Simulation Runs

There are a large number of variables which affect the electrical heating of an oil sand formation. These include the electrical and thermal properties of the oil sand and surrounding formations, the length and thickness of the electrodes, the spacing between electrodes, the thickness of the oil sand, and the rate of heating. In studying the effects of varying one of these variables the other variables must be held fixed.

The initial studies done with MEGAERA were on the effects of different ratios of electrical conductivity of oil sand to that of the surrounding formations, and on the effect of changing the spacing between the electrodes.

4.1 Effects of the Electrical Conductivity Ratio

The author has conducted a series of runs of MEGAERA in which the physical dimensions, thermal properties and heating rates were held fixed, but the electrical conductivities of the oil sand and surrounding formations were varied. The purpose of the series of runs was to determine which of two electrode configurations produces the most favourable temperature distribution for a given ratio between the electrical conductivity of the oil sand and that of the surrounding formations. The problem geometries, labeled configuration A and B, are shown in figures 4.1 and 4.2. In all runs constant power heating of eight kilowatts

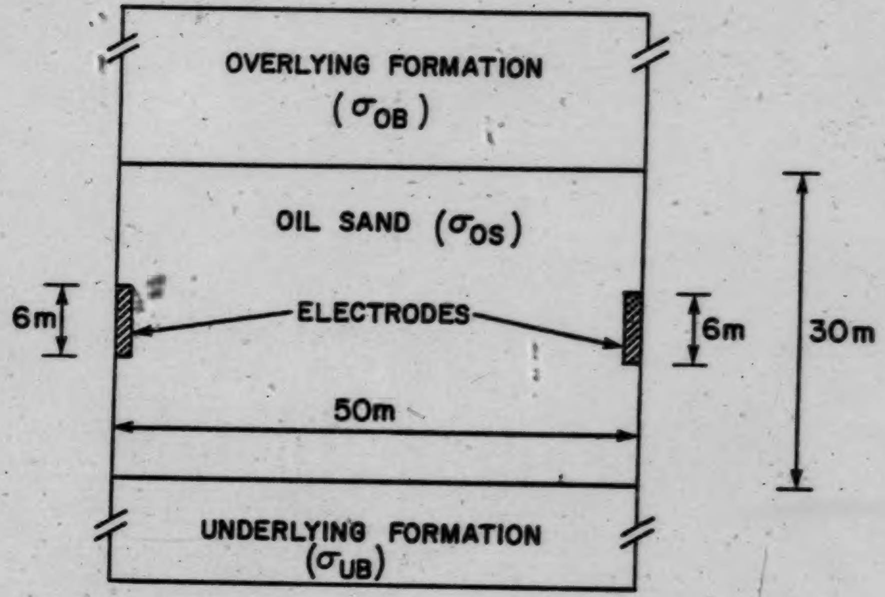

OVERLYING FORMATION
(σ_{OB})
OIL SAND (σ_{OS})
ELECTRODES
6m
6m
30m
50m
UNDERLYING FORMATION
(σ_{UB})

Figure 4.1 Heating Configuration A.

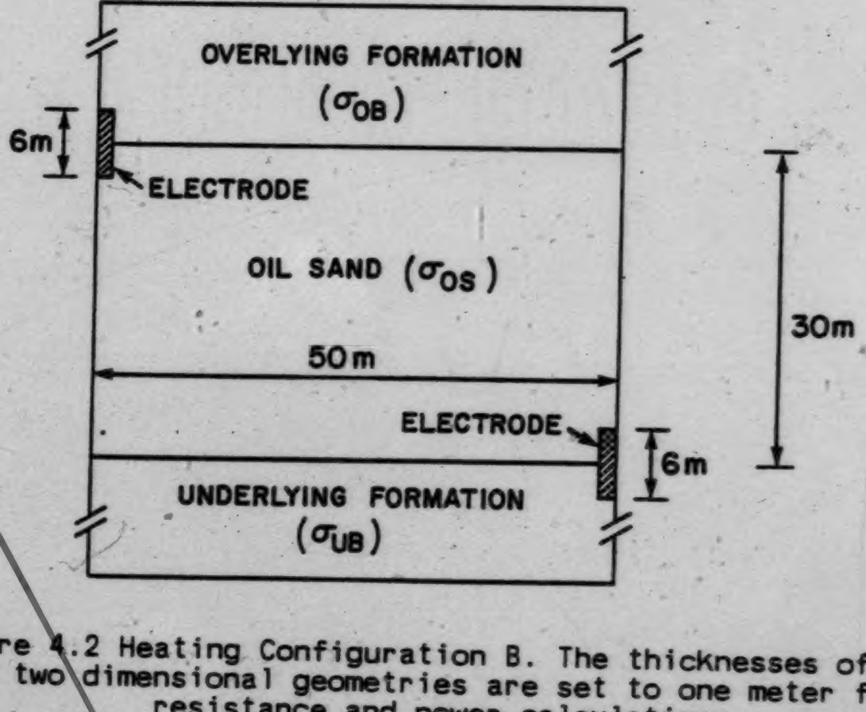

OVERLYING FORMATION
(σ_{OB})
6m
ELECTRODE
OIL SAND (σ_{OS})
30m
50m
ELECTRODE
6m
UNDERLYING FORMATION
(σ_{UB})

Figure 4.2 Heating Configuration B. The thicknesses of these two dimensional geometries are set to one meter for resistance and power calculations.

per meter length of the parallel plate electrodes was applied for a period of one year. Thus, all the runs had the same heating rate regardless of the actual resistance between the electrodes. The results of this series of runs are summarized in Table 4.1 and the temperature plots at the end of the heating period are given in Figures 4.3 to 4.9.

In configuration A (i.e. the electrodes positioned in the middle of the oil sand formation) the maximum temperature, which occurs near the electrodes, increases linearly with the ratio of the electrical conductivities of the surrounding formations to that of the oil sand. (This ratio, electrical conductivity of overburden and underburden to electrical conductivity of oil sand, shall hereafter be refered to as the conductivity ratio.) The temperature in the formation mid way between the electrodes dropped as the conductivity ratio increased. Configuration B had the opposite trend with the electrode temperature dropping and the mid formation temperature rising with increased conductivity ratio. For conductivity ratios greater than two configuration B had a more desirable temperature profile than configuration A. A graph of temperature vs. conductivity ratio is given in figure 4.10.

When the fraction of the total energy input that is retained in the oil sand formation is computed, configuration A is superior to configuration B even with the conductivity ratio as high as five. With a conductivity ratio of one half, configuration A heating will retain about

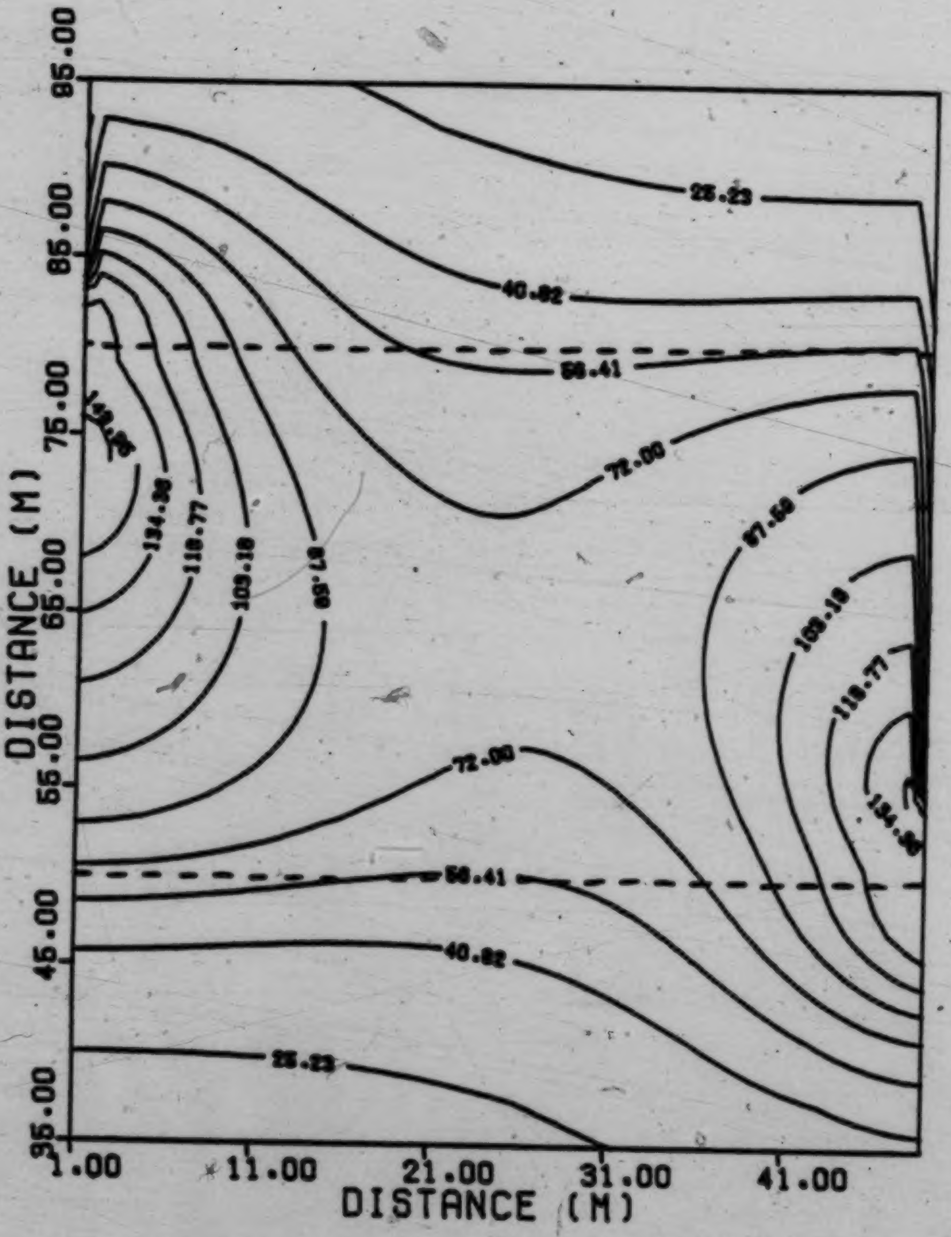


Figure 4.3 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 10, configuration B, with an electrical conductivity ratio (overburden, underburden/ oil sand) of 10.

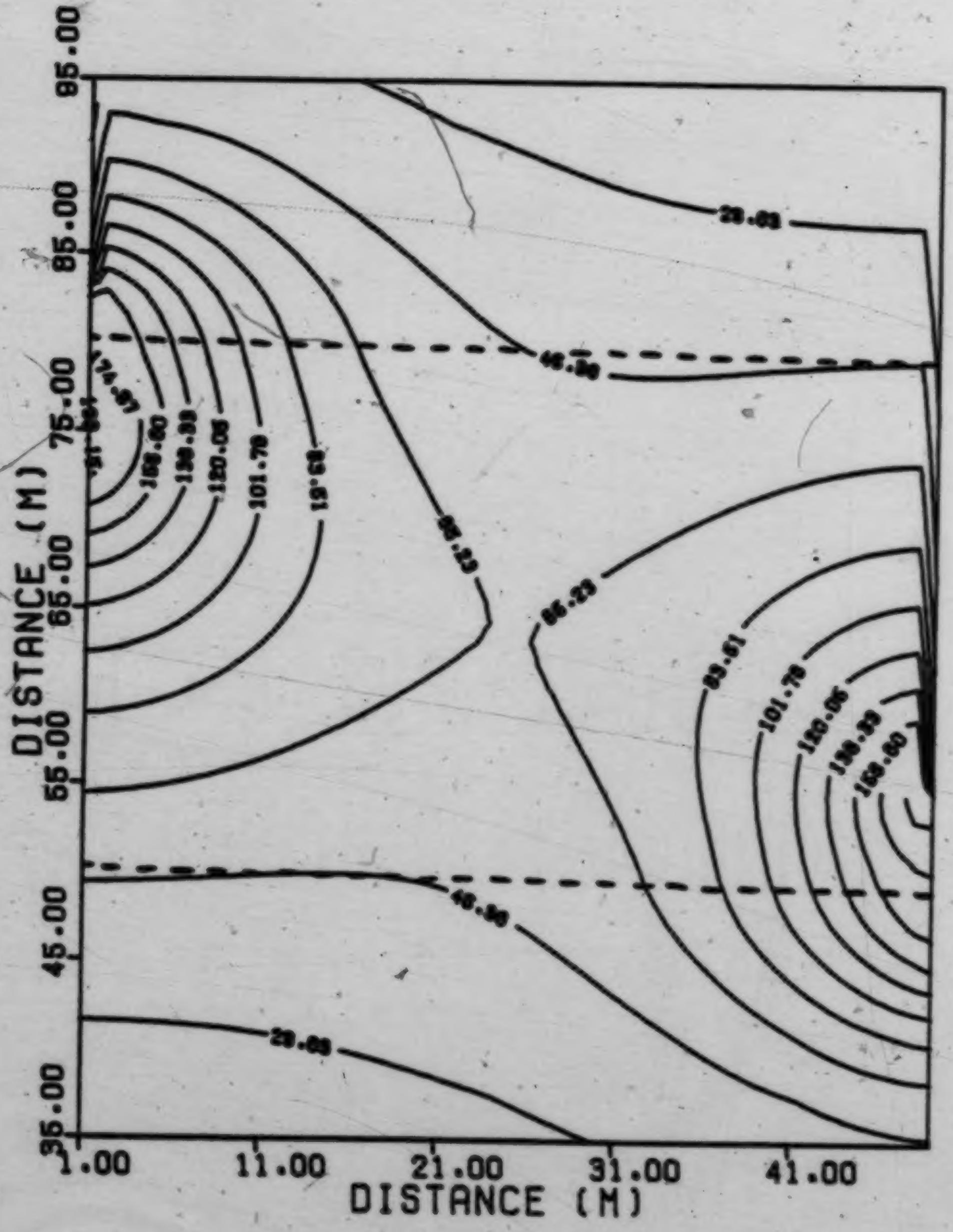


Figure 4.4 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 12, configuration B, with an electrical conductivity ratio (overburden, underburden/ oil sand) of 2.

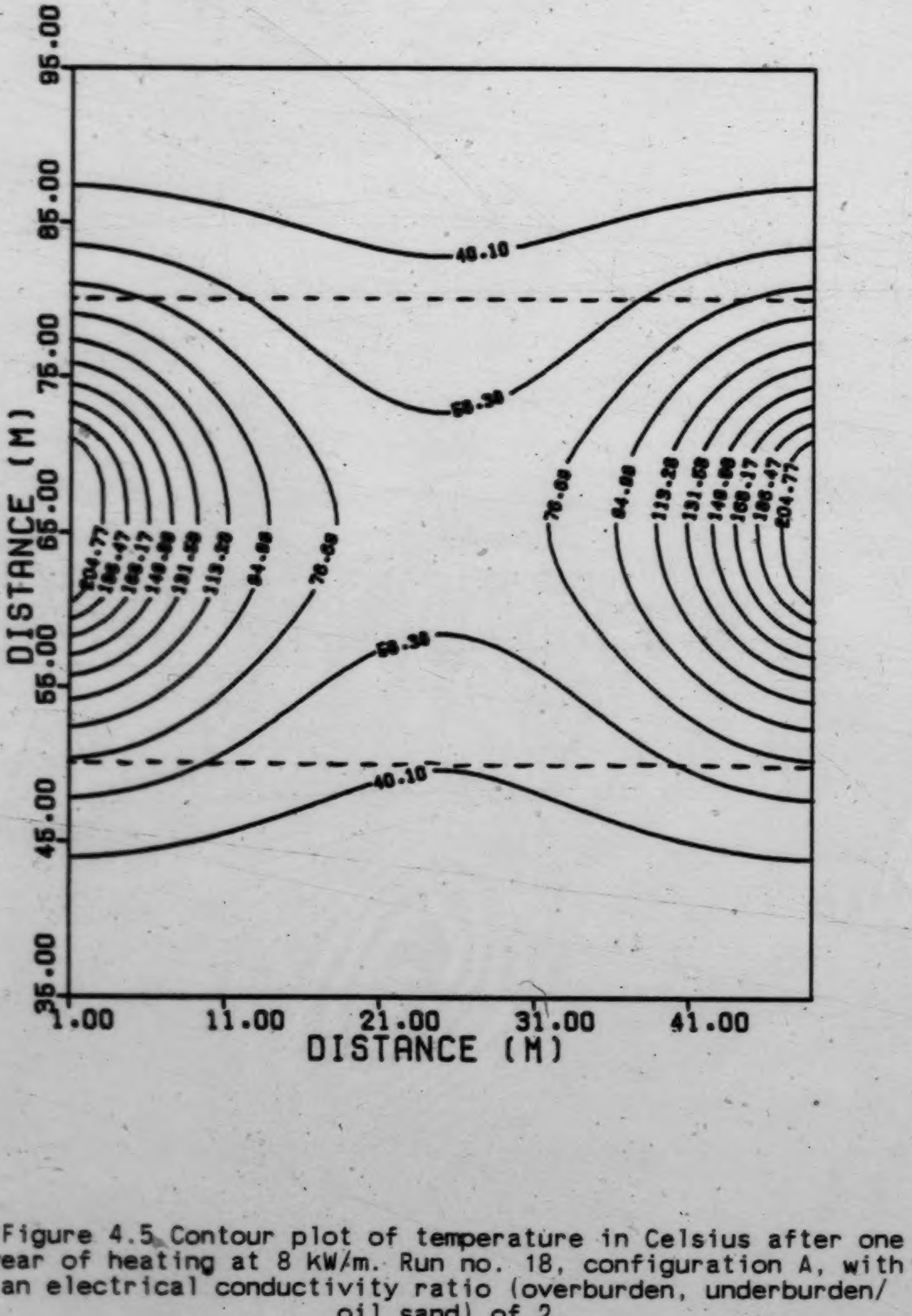


Figure 4.5 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 18, configuration A, with an electrical conductivity ratio (overburden, underburden/oil sand) of 2.

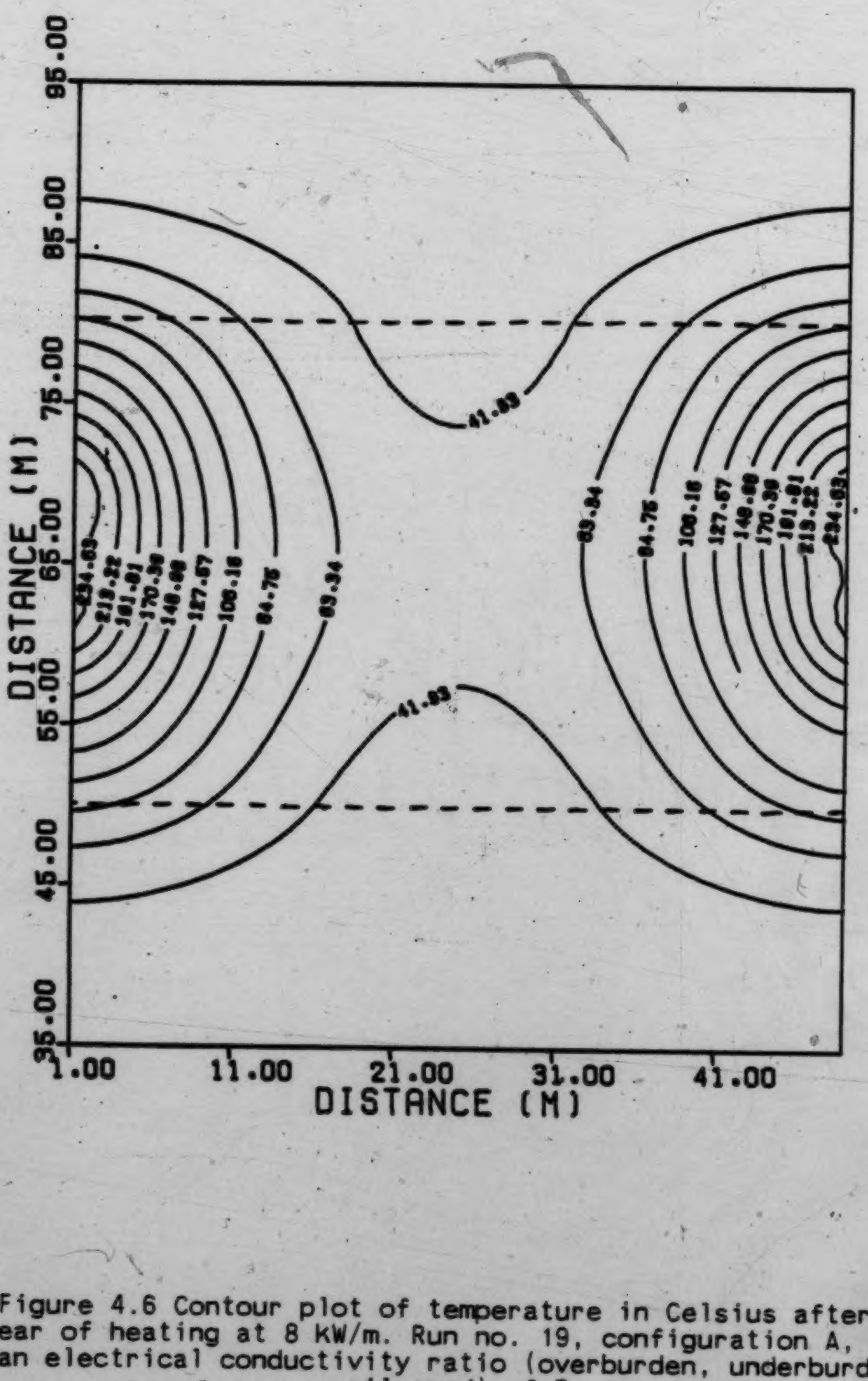


Figure 4.6 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 19, configuration A, with an electrical conductivity ratio (overburden, underburden/ oil sand) of 5.

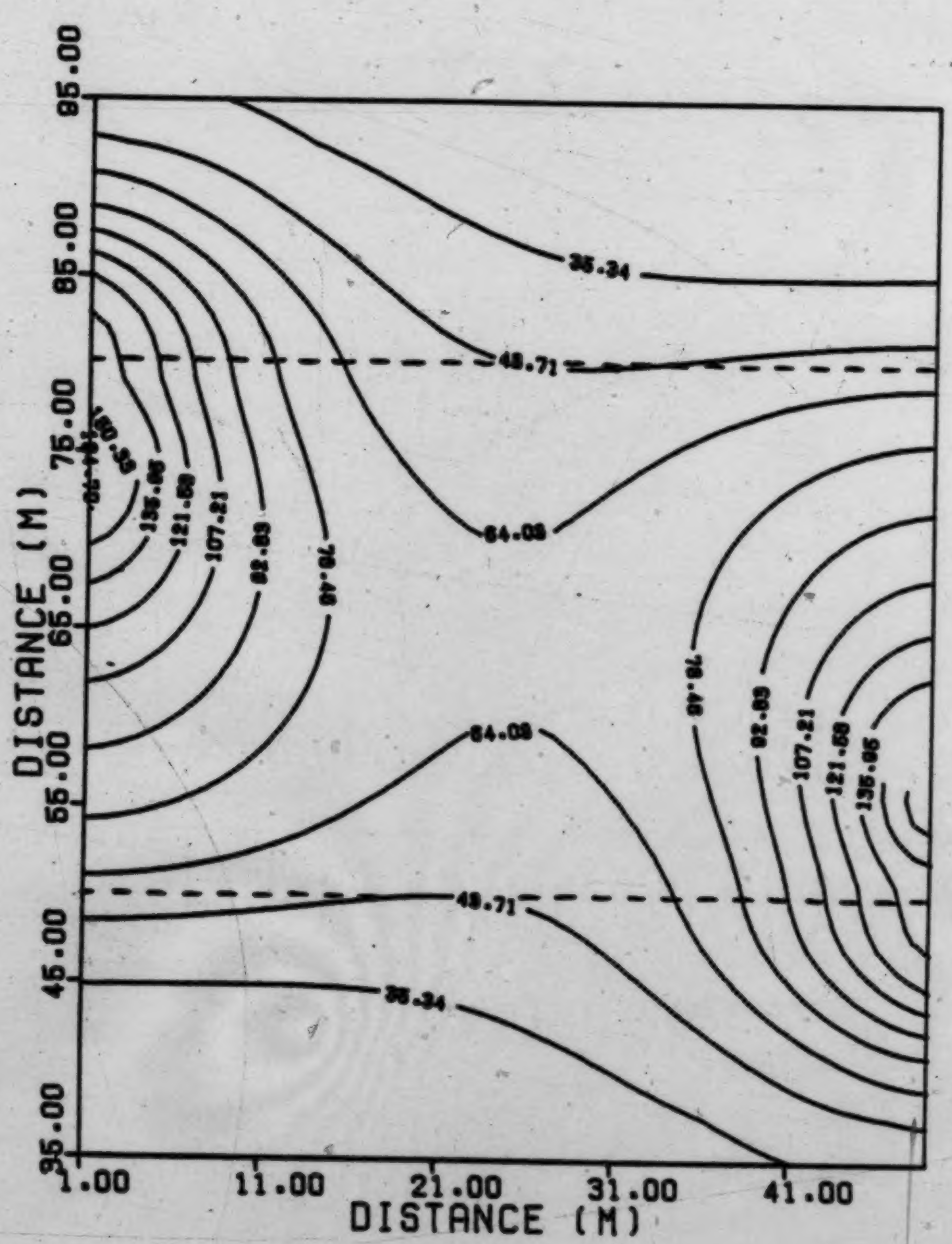


Figure 4.7 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 20, configuration B, with an electrical conductivity ratio (overburden, underburden/oil sand) of 5.

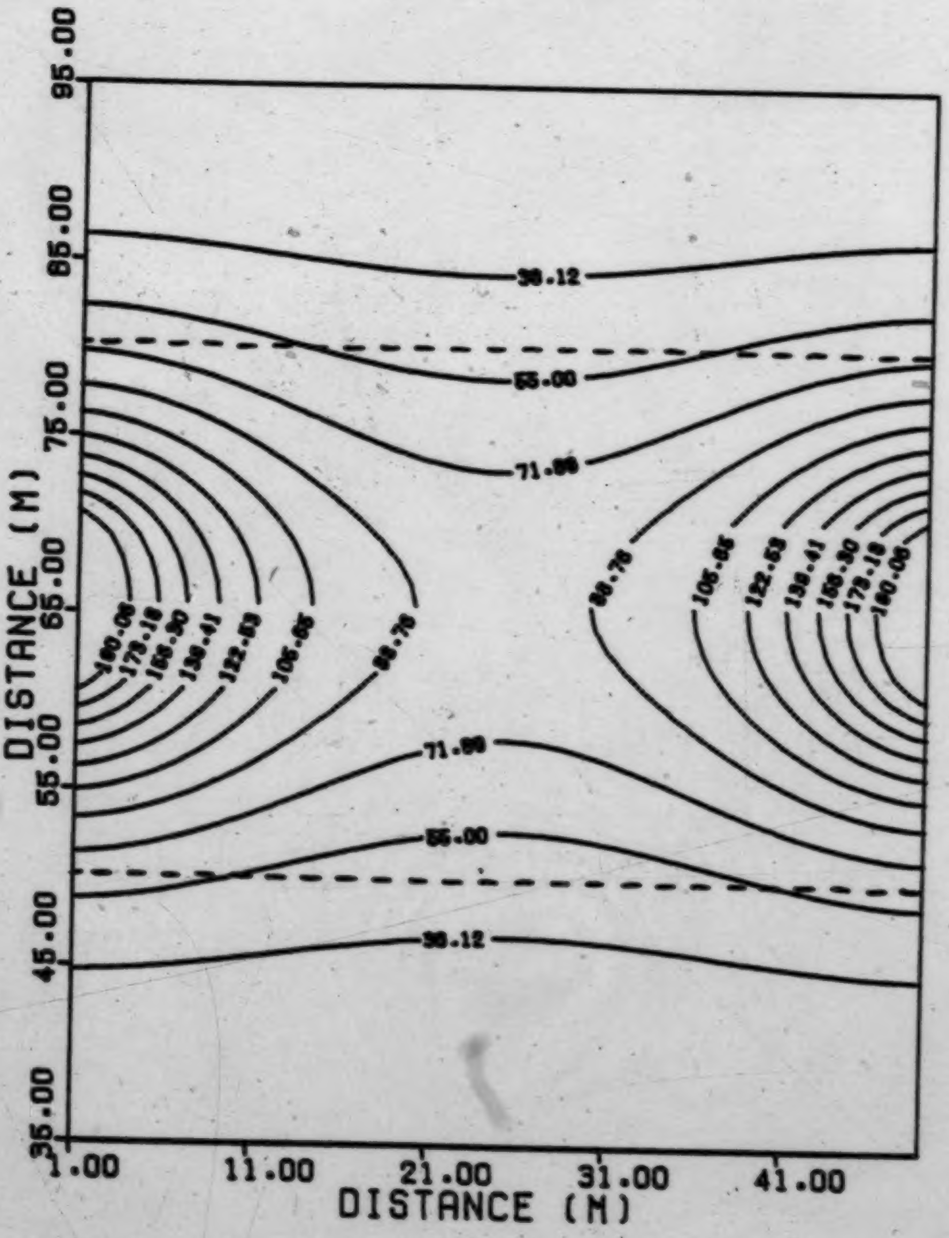


Figure 4.8 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 21, configuration A, with an electrical conductivity ratio (overburden, underburden/oil sand) of 1.

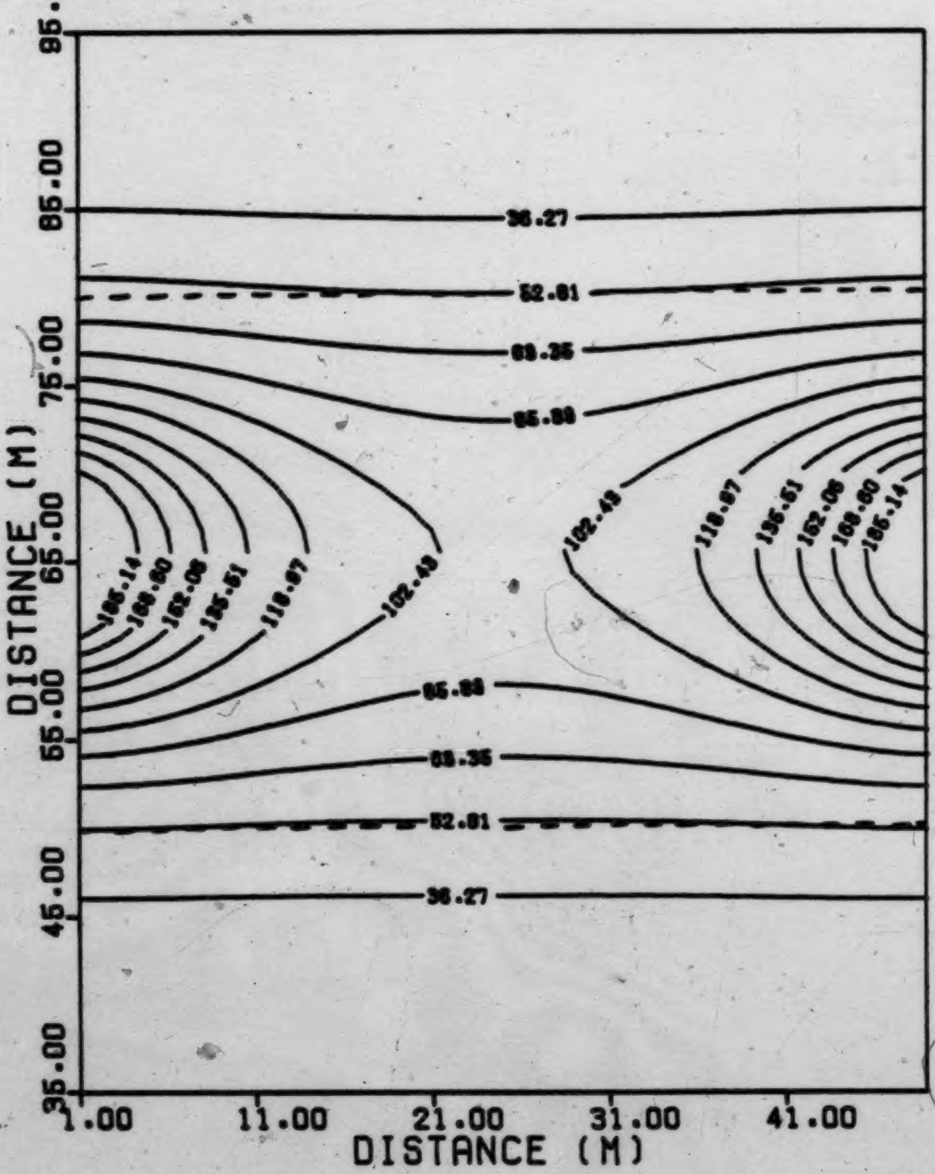


Figure 4.9 Contour plot of temperature in Celsius after one year of heating at 8 kW/m. Run no. 22, configuration A, with an electrical conductivity ratio (overburden, underburden/ oil sand) of .5

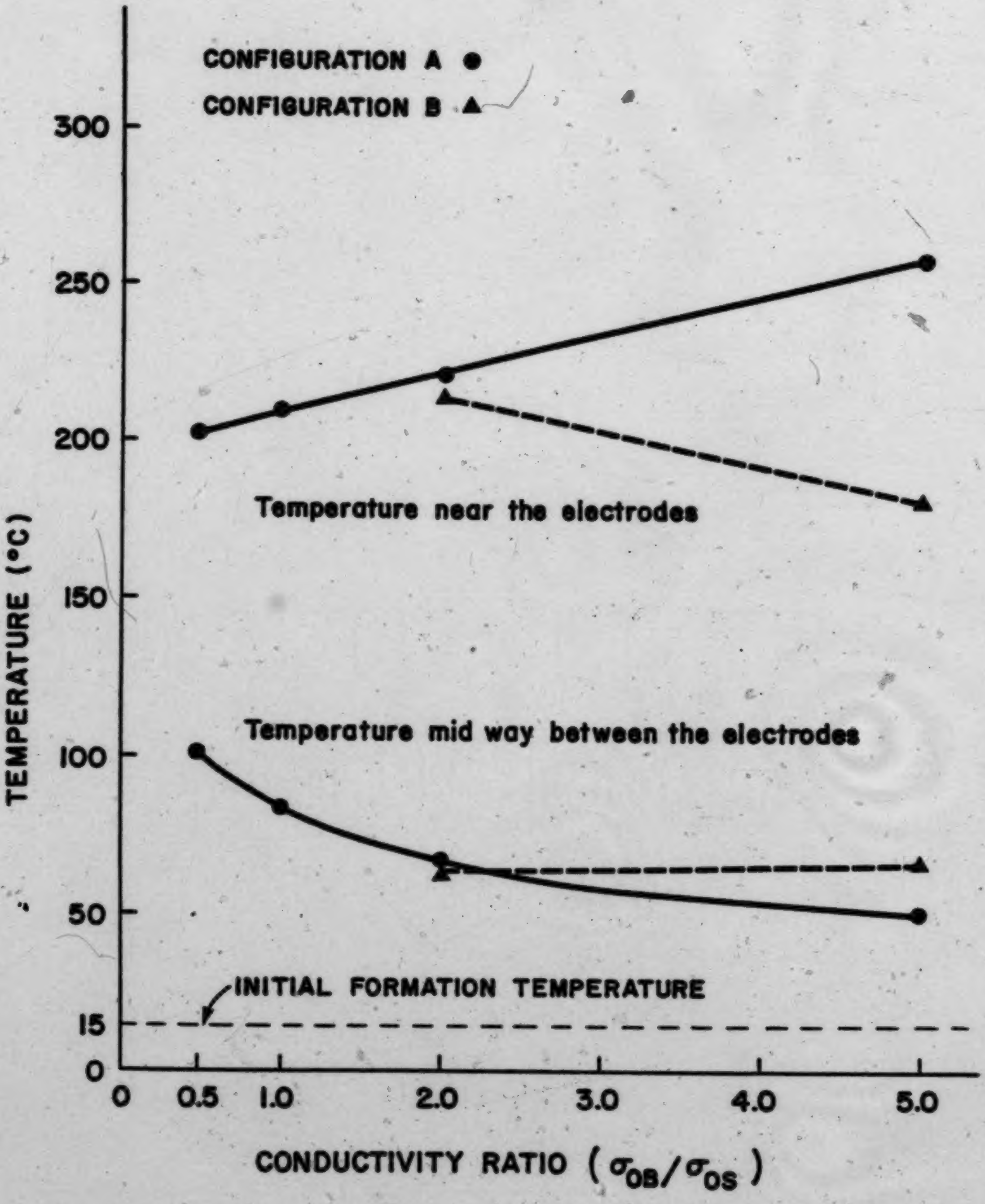


Figure 4.10 Temperature vs. the ratio of the electrical conductivities of the surrounding formations to that of the oil sand.

Table 4.1 Temperature rise and thermal efficency as a function of the relative conductivity of the oil sand and surrounding formations.

The conductivity of the oil sand is 1.0 X 10^{-3} S/m for all runs. Constant power heating of 8 kW per meter was applied for one year. Initial reservoir temperature is 15°C.

1. Run No.	2. Configuration	3. σ_{ob} S/m x10^{-3}	4. σ_{ub} S/m x10^{-3}	5. Max. Temp. °C	6. Mid Temp. °C	7. Heating in Oil Sand %	8. Energy in Oil Sand %
22	A	0.5	0.5	202	101	88	81
21	A	1.0	1.0	210	85	84	78
18	A	2.0	2.0	220	67	81	75
12	B	2.0	2.0	212	65	65	62
19	A	5.0	5.0	257	50	84	75
20	B	5.0	5.0	180	66	68	65
10	B	10.	10.	177	75	74	69

1. Runs were numbered in chronological order.
2. Dimensions and electrode positions are given in figures 4.1 and 4.2.
3. Conductivity of overlying formation.
4. Conductivity of underlying formation.
5. Maximum temperature near the electrodes.
6. Maximum temperature mid way between the electrodes.
7. Electrical energy dissipated in the oil sand formation as a percentage of the total energy dissipated in the oil sand and surrounding formations.
8. The energy stored in the oil sand formation at the end of one year, as a percentage of the total energy input. The difference between columns 7 and 8 is due to thermal conduction.

eighty percent of the energy in the oil sand after heating for one year. When the conductivity ratio is increased to five, configuration A will retain three quarters of the energy in the oil sand formation, while heating with configuration B will result in only sixty five percent of the energy input remaining in the oil sand at the end of the year. In both configurations most of the energy is lost by current passing through (and thus heating) the overlying and underlying formations. Thermal conduction typically accounts for the loss of five to ten percent of the total energy input, or ten to forty percent of the total energy lost to the surrounding formations.

When heating in configuration A, the conductivity ratio determines what fraction of the current enters the surrounding formations. As the conductivity ratio increases more of the current travels most of the distance between the electrodes in the overlying or underlying formations, where relatively low resistance is encountered. As a result the bulk of the voltage drop and the heating is in the oil sand near the electrodes, where the current is travelling from the electrode almost directly into the surrounding formations. This leads to a higher electrode temperature and a lower temperature midway between the electrodes. When heating in configuration B, a high conductivity ratio results in the overlying and underlying formations acting like large extended electrodes. Most of the voltage drop is across the oil sand formation from the overlying formation to the underlying formation. However, in configuration B most of the current must travel some of the distance between the electrodes through the surrounding formations and this results in a lower fraction of the total heating occuring in the oil sand formation. This explains why, for a conductivity ratio of five, configuration B has a more uniform temperature profile in the oil sand, while configuration A has more of the energy input retained in the oil sand.

The total oil sand volume in both configurations A and B, per meter thick slice of the formation, is 1500 m^3. The chemical energy content of oil sand is of the order of 10^{10} Joules/m^3, so the total chemical energy in the heated formation is about 10^{13} Joules. An electrical preheat of 8 kW/m for one year uses about 2.5 X 10^{11} Joules of electrical energy, which is of the order of one percent of the chemical energy in place.

4.2 Effects of changing the distance between the electrodes

Computer runs were carried out with 75 m and 100 m spacings between the electrodes, with the other dimensions and electrode positions as in configuration A (figure 4.1). The power input was increased to 12 kW/m for the 75 m spacing and to 16 kW/m for the 100 m spacing, but the heating time remained at one year. The results of these runs, and similar runs done with 50 m spacings, are given in table 4.2. The temperature contour plots for the end of the year of heating are given in figures 4.11 to 4.17.

Even though the power output from the electrodes is increased in proportion to the distance between the electrodes, the maximum temperature near the electrodes remained constant for conductivity ratios of one half and one. For the runs with different electrical conductivities for the overlying and underlying formations, the maximum temperature near the electrodes increased by fifteen percent when the distance between the electrodes was increased from 50 m to 100 m. The temperature midway between the electrodes decreased with an increase in the distance between the electrodes. A graph of the temperature midway between the electrodes vs. the distance between the electrodes is given in figure 4.18. The fraction of the total energy input that is retained in the oil sand formation at the end of the year also declines with an increase in the distance between the electrodes.

As the distance between the electrodes increases, more of the current reaches the surrounding formations, and less desirable temperature profiles and thermal efficencies result. A tradeoff must be made between the cost of the extra electrodes for a closely spaced heating configuration and the extra electrical energy and heating time required to obtain the same temperature profile for a larger spacing. The effect of changing the distance between the electrodes is more pronounced for higher conductivity ratios. The optimal distance between the electrodes will be a function

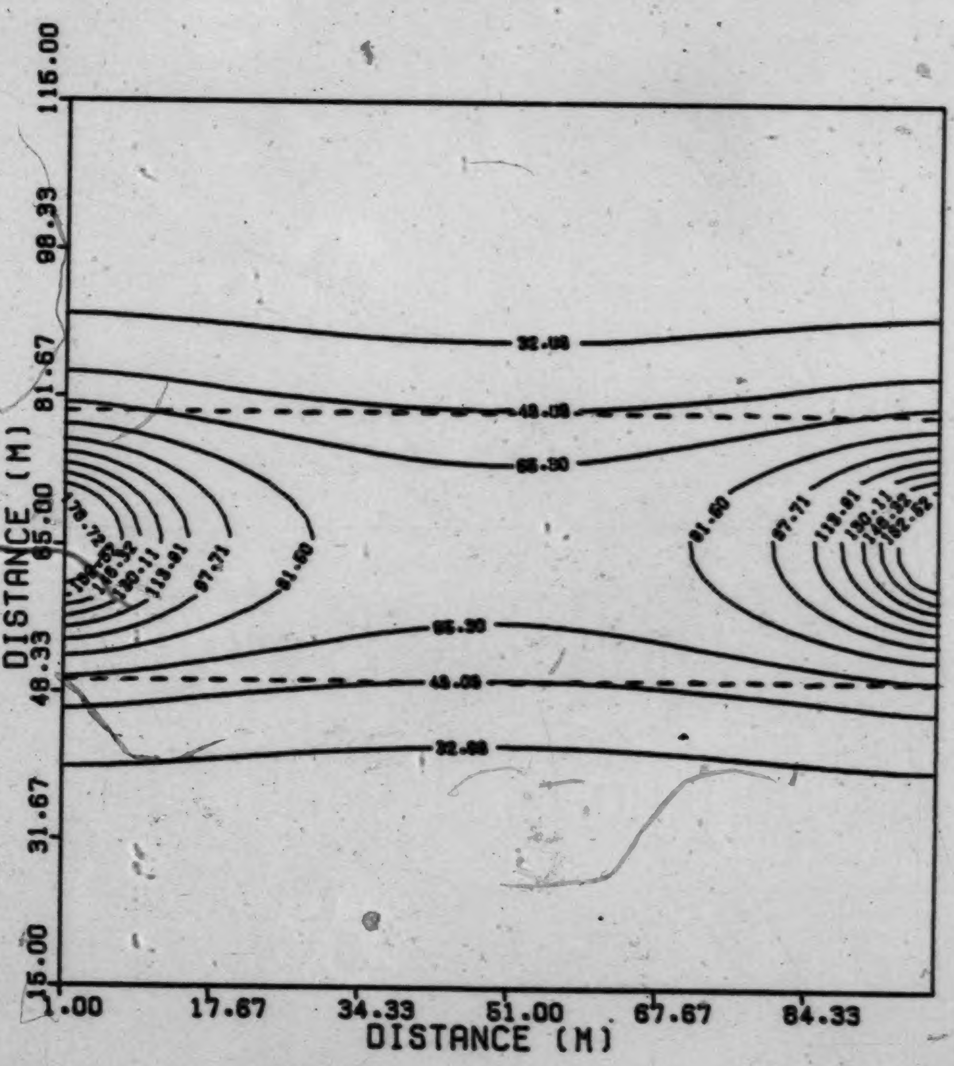


Figure 4.11 Contour plot of temperature in Celsius at the end of heating for one year at 16 kW/m. Run no. 23, configuration A, with a conductivity ratio (overburden, underburden/ oil sand) of .5

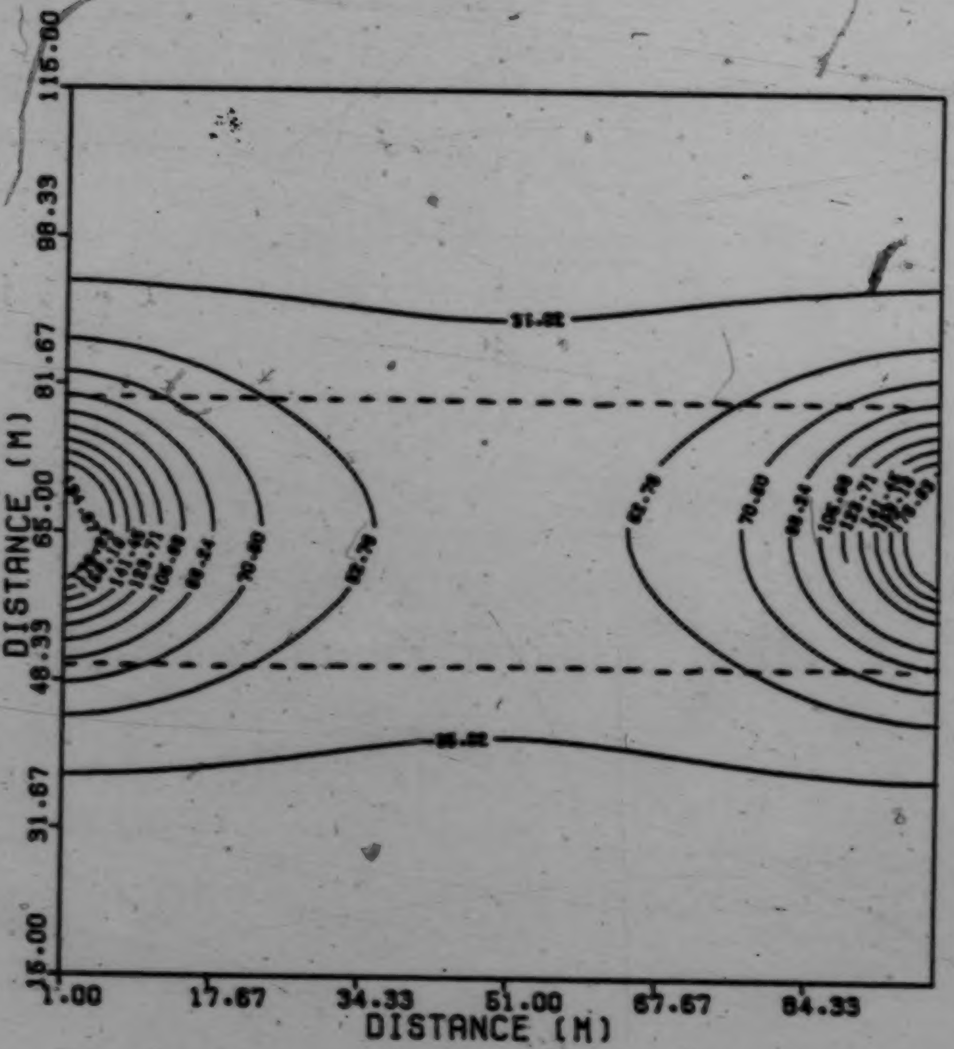


Figure 4.12 Contour plot of temperature in Celsius at the end of heating for one year at 16 kW/m. Run no. 24, configuration A, with a conductivity ratio (overburden, underburden/ oil sand) of 1.

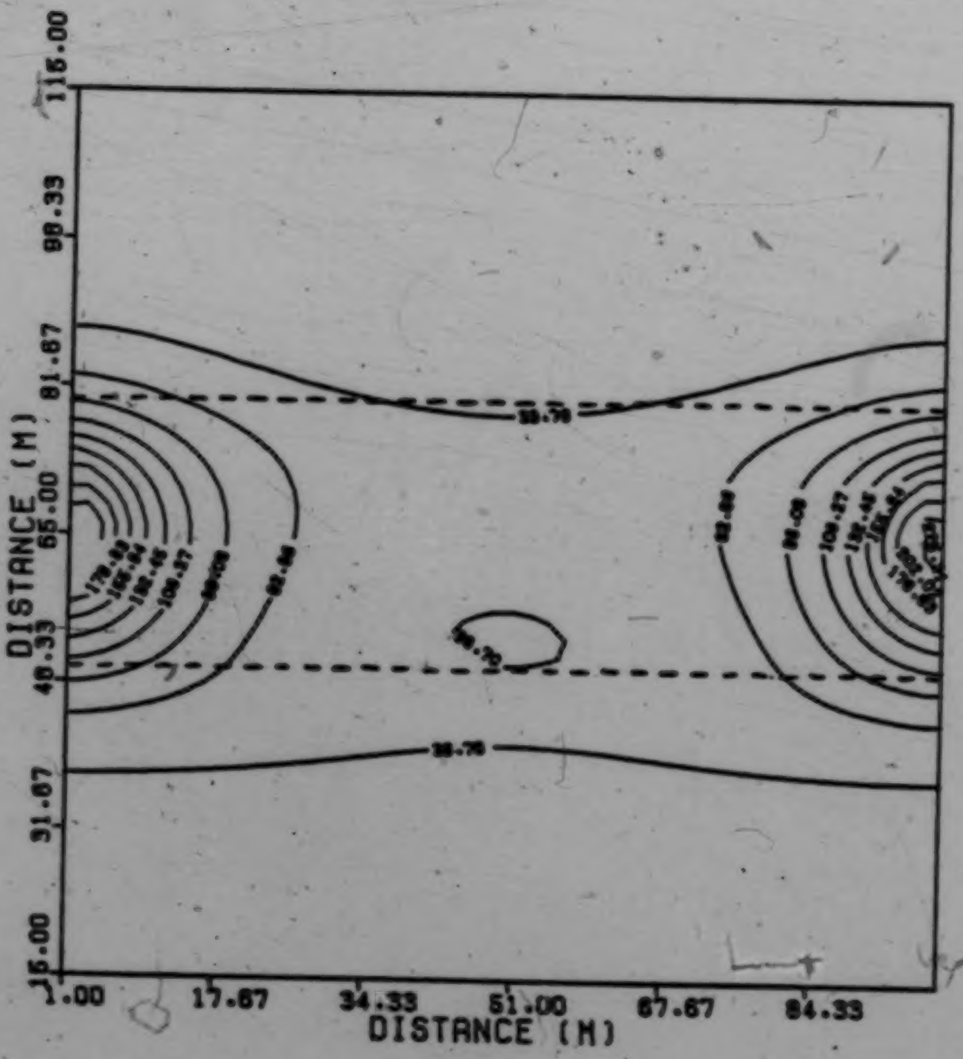


Figure 4.13 Contour plot of temperature in Celsius at the end of heating for one year at 16 kW/m. Run no. 25, overburden half as conductive and underburden twice as conductive as oil sand.

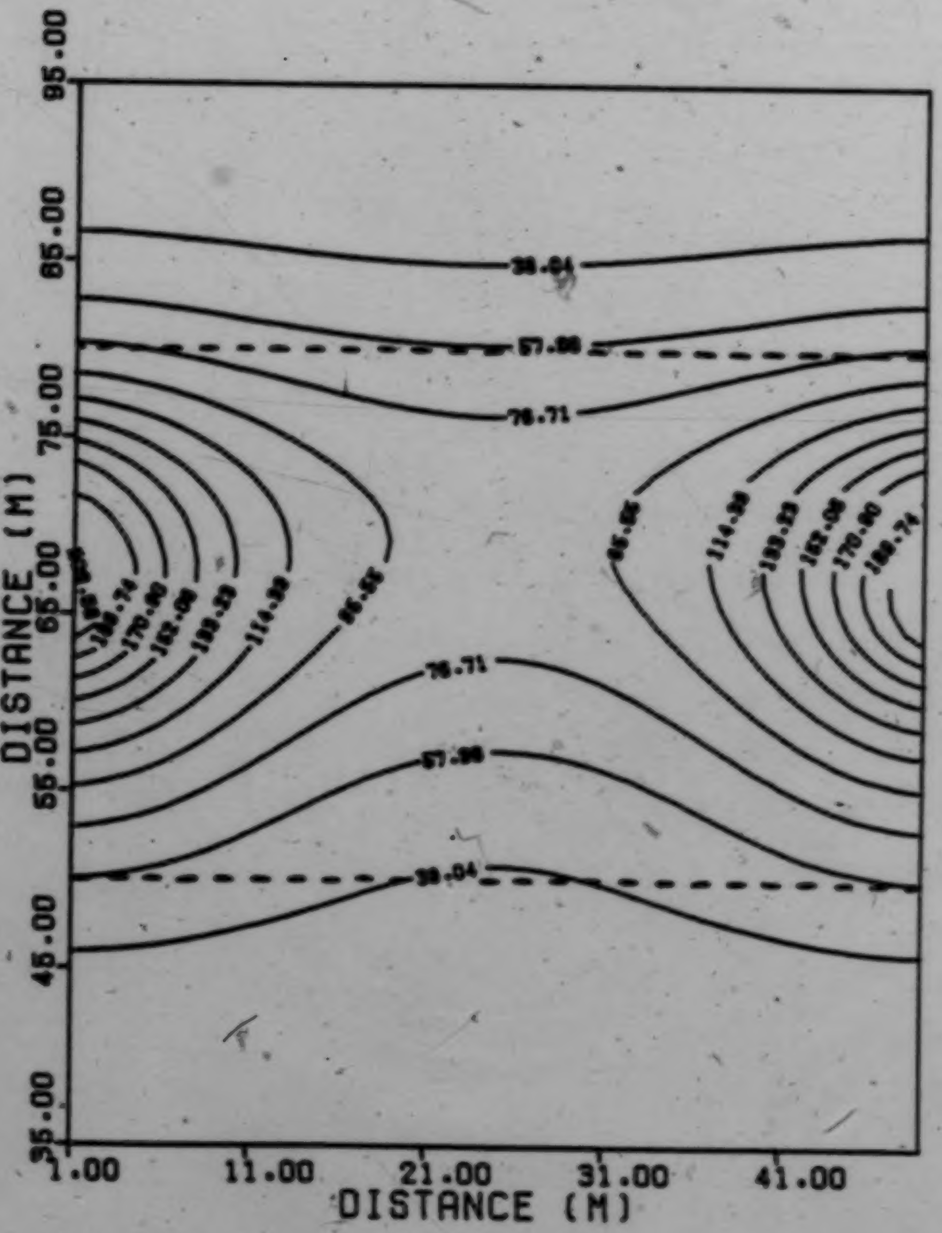


Figure 4.14 Contour plot of temperature in Celsius at the end of heating for one year at 8 kW/m. Run no. 26, overburden half as conductive and underburden twice as conductive as oil sand.

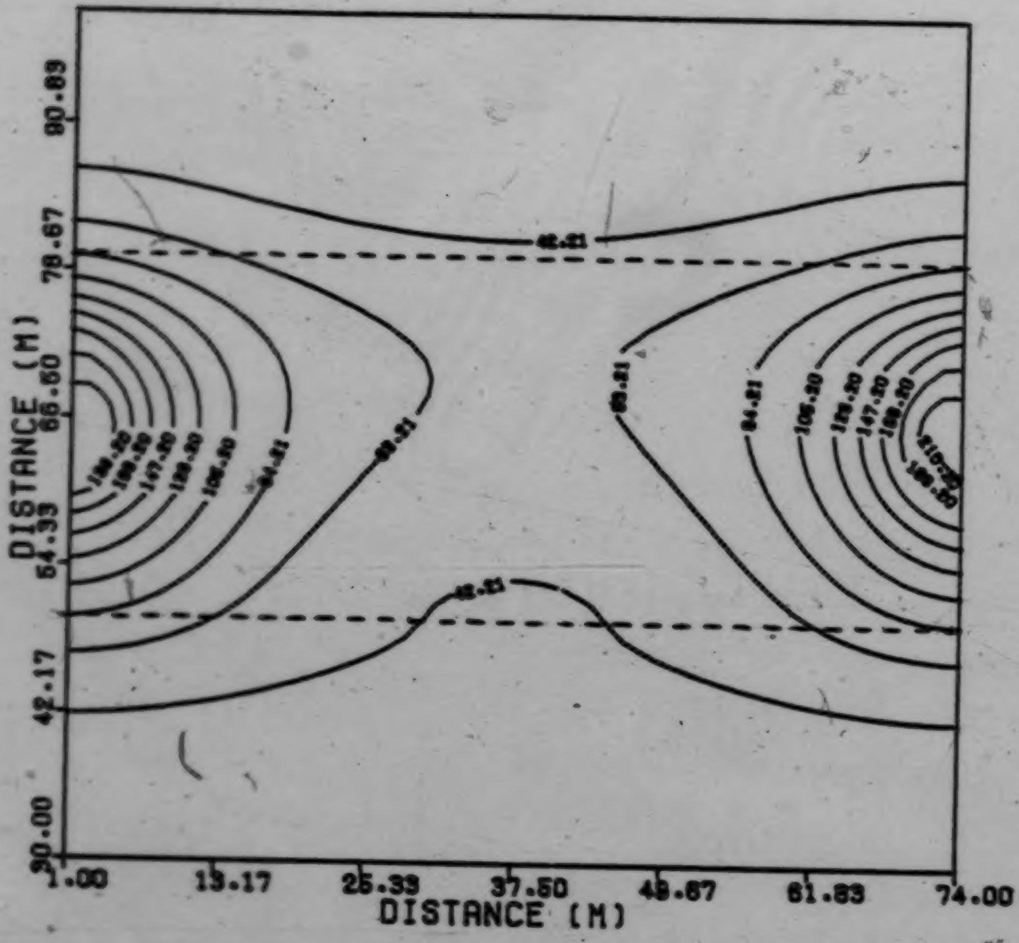


Figure 4.15 Contour plot of temperature in Celsius at the end of heating for one year at 12 kW/m. Run no. 32, overburden half as conductive and underburden twice as conductive as oil sand.

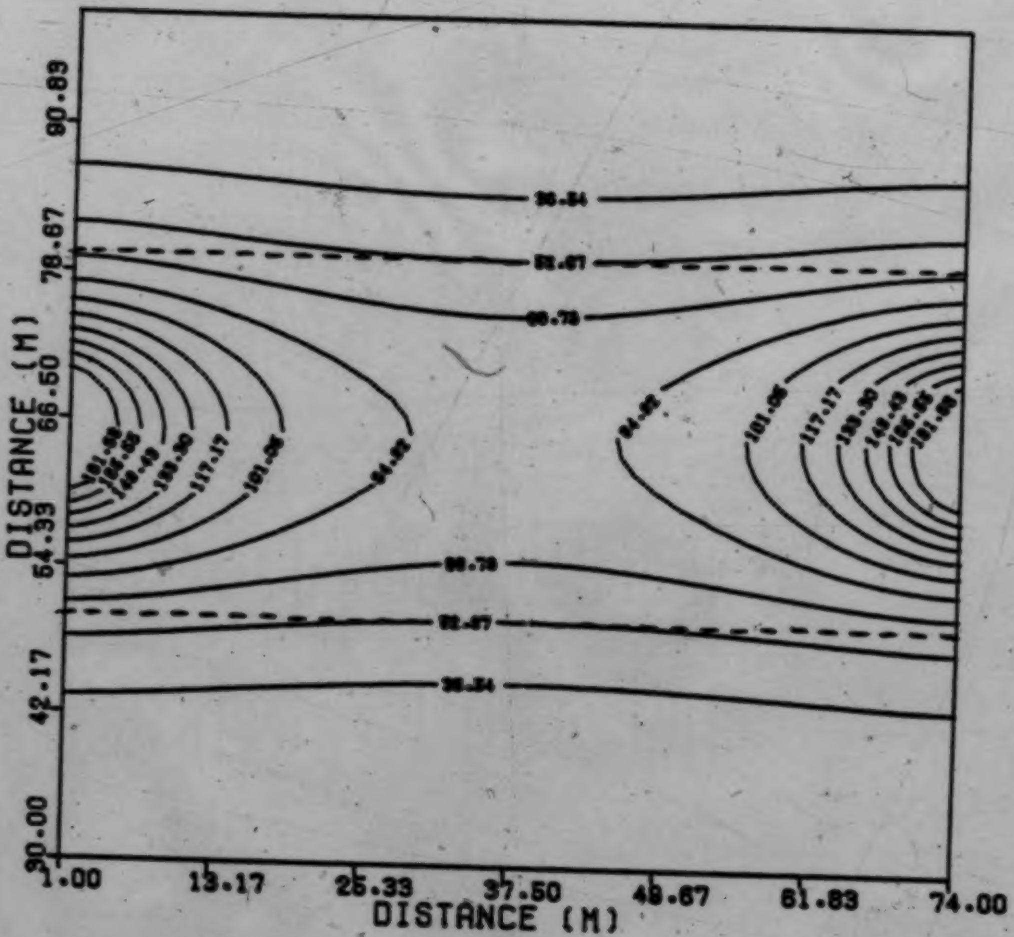


Figure 4.16 Contour plot of temperature in Celsius at the end of heating for one year at 12 kW/m. Run no. 34, configuration A, with a conductivity ratio (overburden, underburden/ oil sand) of .5

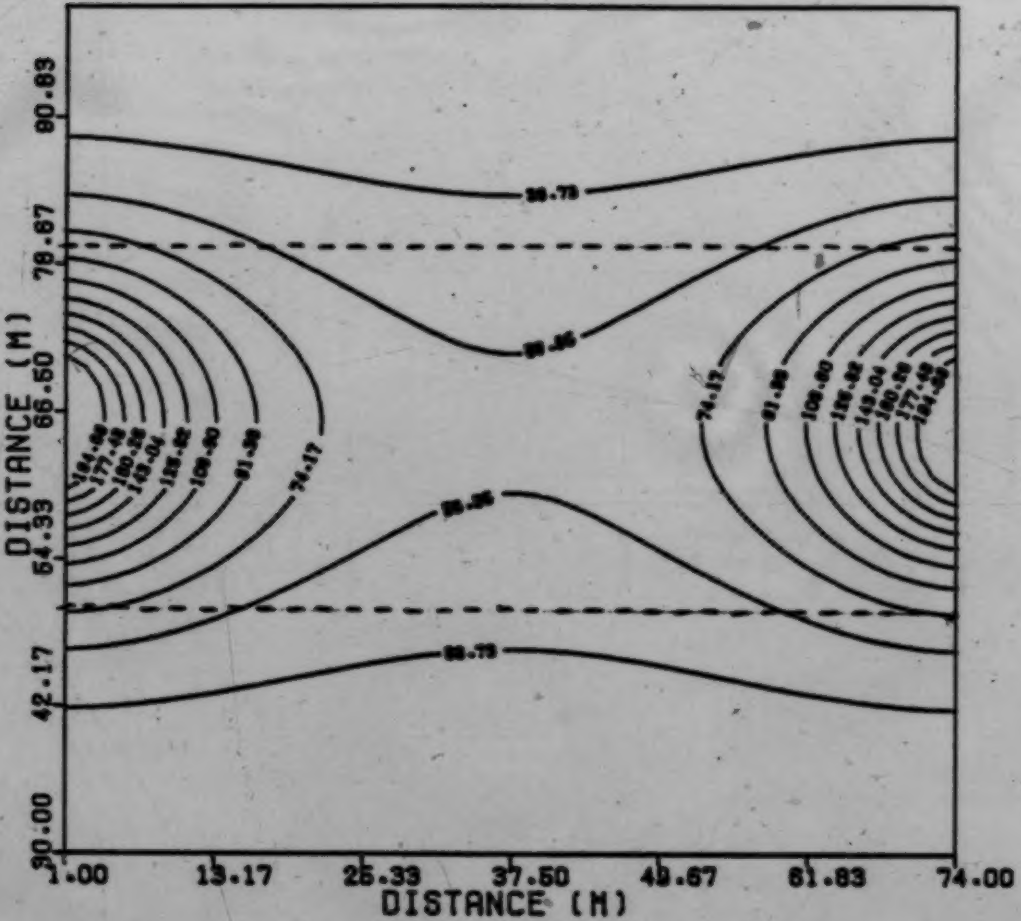
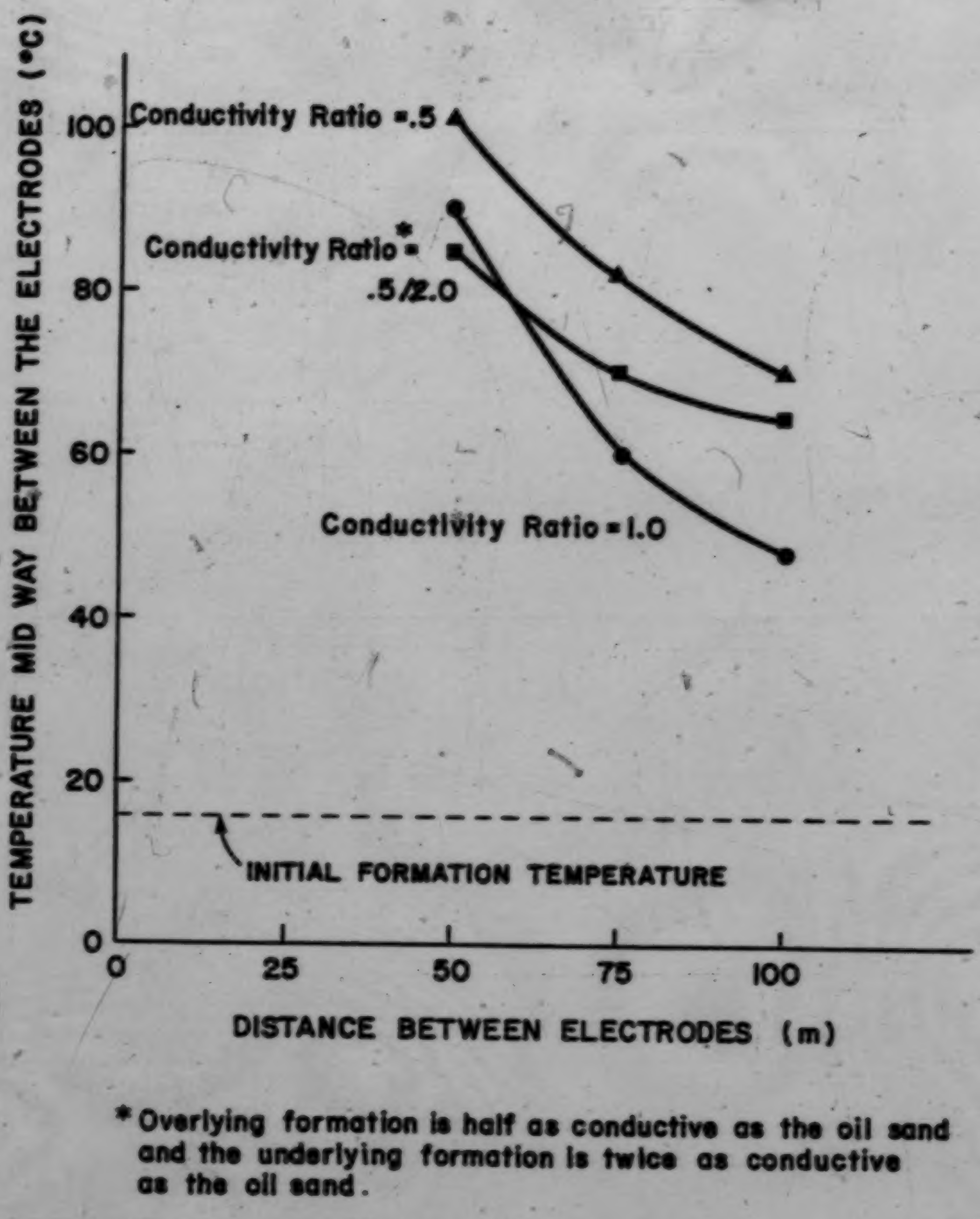


Figure 4.17 Contour plot of temperature in Celsius at the end of heating for one year at 12 kW/m. Run no. 35, configuration A,.with a conductivity ratio (overburden, underburden/ oil sand) of 1.



* Overlying formation is half as conductive as the oil sand and the underlying formation is twice as conductive as the oil sand.

Figure 4.18 Temperature midway between the electrodes vs. the distance between the electrodes. Electrode positions are as in configuration A, except for the mixed conductivity ratio where the electrodes were moved up three meters.

Table 4.2 Temperature rise and thermal efficency as a function of distance between the electrodes.

The conductivity of the oil sand is 1.0 X 10^{-3} S/m for all runs. Constant power heating of 8 kW per meter in the runs with a 50 m spacing, 12 kW per meter in the runs with a 75 m spacing, and 16 kW per meter in the runs with a 100 m spacing was applied for one year. Formation thicknesses and electrode sizes and positions are as in configuration A (Fig. 4.1).

1. Run No.	2. Spacing m	3. σ_{ob} S/m x10^{-3}	4. σ_{ub} S/m x10^{-3}	5. Max. Temp. °C	6. Mid Temp. °C	7. Heating in Oil Sand %	8. Energy in Oil Sand %
22	50	0.5	0.5	202	101	88	81
34	75	0.5	0.5	196	82	79	73
23	100	0.5	0.5	195	72	72	67
21	50	1.0	1.0	210	85	84	78
35	75	1.0	1.0	210	60	71	66
24	100	1.0	1.0	210	48	62	58
26	50	0.5	2.0	230	90	85	79
32	75	0.5	2.0	258	74	74	68
25	100	0.5	2.0	264	46	65	60

1. Runs were numbered in chronological order.
2. Distance between the electrodes.
3. Conductivity of overlying formation.
4. Conductivity of underlying formation.
5. Maximum temperature near the electrodes.
6. Maximum temperature mid way between the electrodes.
7. Electrical energy dissipated in the oil sand formation as a percentage of the total energy dissipated in the oil sand and surrounding formations.
8. The energy stored in the oil sand formation at the end of one year, as a percentage of the total energy input. The difference between columns 7 and 8 is due to thermal conduction.

of many parameters, with two important factors being the cost of the wells and electrodes and the relative conductivities of the surrounding formations to the oil sand.

5. Conclusions

The results of the full scale simulation runs done using the computer program MEGAERA are encouraging. These results indicate that, by using less than five percent of the chemical energy in place in a typical thickness Athabasca oil sand formation, it is possible to raise the temperature midway between the electrodes to 80°C, with higher temperatures near the electrodes. The viscosity of the bitumen (Athabasca oil sand) would be about 1000 cp midway between the electrodes, and considerably less near the electrodes. This is low enough to allow an efficent steam drive to be conducted.

The comparisons done between the analytic solution and the computer simulation indicate that the finite difference scheme used was correct. The convergence error due to the explicit coupling of the electrical and thermal equations remains, but, if the maximum temperature change per timestep is kept less than five percent, this error will not be significant.

The best test of both the mathematical model and the finite difference scheme that were used is conducted by comparing the simulator results with a physical model run. When this was done the simulator results were in agreement with the measured temperatures from the physical model run, except in the immediate vicinity of the electrodes. Thus, at least for temperatures up to 90°C (above which accurate electrical conductivity data is not available), the mathematical model used will accurately predict temperatures and resistance for electrical heating of solid materials.

There are several items which should be kept in mind when interpreting the results of the full scale simulator runs. First, the electrical and thermal properties of oil sand have not been reported in the liturature for temperatures above 90°C. In the simulator the thermal properties were assumed independent of temperature, and the electrical conductivity was assumed to increase linearly with temperature, up to 250°C. Second, the conductivity of the oil sand near the electrodes in a field test would be altered by the injection of brine near the electrode. These two factors may result in significant error in the predicted temperatures near the electrodes. Finally, the conductivity of the oil sand was made uniform throughout the formation for the simulation runs. This was by choice and not due to any limitation of the simulator. In an actual oil sand formation there is probably variations in electrical conductivity with height in the formation and due to the presence of shale breaks and clay deposits.

The investigation to date indicates that the electrical preheat method may be developed into an economical method of preparing an oil sand formation for a steam flood or fire flood. Further measurement of the electrical and thermal properties of oil sand need to be performed at formation pressures and temperatures in the range of 100°C to 300°C. The computer simulator should be extended to three dimensions so that simulation of five spot patterns are possible in addition to the parallel plate electrode (or line drive) geometry which was studied in this thesis. Finally, due to variations in the conductivities of the oil sand and the surrounding formations no one electrode configuration is optimal and each recovery location will have to be investigated individually to determine the optimal electrode size, positioning and spacing.

References and Bibliography

[1] F.W. Camp, **The Tar Sands of Alberta, Canada**, Third Edition, Cameron Engineering, Inc., Denver, Colorado, 1976, 77 pp.

[2] C.P. Outtrim, R.G. Evans, Alberta's Oil Sands Reserves and their Evaluation, **The Oil Sands of Canada-Venezuela 1977**, CIM Special Vol. 17, pp. 36

[3] J.J.Fitzgerald, **Black Gold with Grit**, Grey's Publishing Ltd., Sidney, B.C., 1978, 221 pp.

[4] J.H. Nicholls, R.W. Luhning, Heavy Oil Sand and Insitu Pilot Plants in Alberta (Past and Present), **The Oil Sands of Canada-Venezuela 1977**, CIM Special Vol. 17, pp. 528

[5] Anon., Thermal Recovery System Uses Electricity, **Petroleum Engineering**, July 1969, pp. 44

[6] S.M. Farouq Ali, D.A. Redford, Physical Modeling of In Situ Recovery Methods for Oil Sands, **The Oil Sands of Canada-Venezuela 1977**, CIM Special Vol. 17, pp.319

[7] F.E. Vermeulen, F.S. Chute, M.R. Cervenan, Physical Modeling of the electromagnetic heating of oil sand and other earth-type and biological materials, **Can. Elec. Eng. J.**, vol. 4, no. 4, 1979, pp. 19

[8] J.C. Todd, E.P. Howell, Numerical Simulation of In Situ Electrical Heating to Increase Oil Modility, **The Oil Sands of Canada-Venezuela 1977**, CIM Special Vol. 17, pp. 477

[9] S.A. El-Feky, Theoretical and Experimental Investigation of Oil Recovery by the Electrothermic Technique, Ph.D. Disertation, University of Missouri-Rolla, 1977.

[10] F.S. Chute, F.E. Vermeulen, M.R. Cervenan, F.J.

McVea, Electrical Properties of Athabasca Oil Sands, Can. J. Earth Sci., Vol 16, No. 10, 1979, pp. 2009

[11] M.R. Cervenan, F.E. Vermeulen, F.S. Chute, Thermal Conductivity and Specific Heat of Oil Sand Samples, presented to the Applied Oilsands Geotechnical conference, June 1980

[12] S. Ramo, J.R. Whinnery, T. Van Duzer, **Fields and Waves in Communication Electronics**, John Wiley and Sons Ltd., 1965, pp. 234

[13] L.M. Magid, **Electromagnetic Fields, Energy, and Waves**, John Wiley and Sons Ltd., 1972, Chap. 9

[14] J.D. Jackson, **Classical Electromagnetics**, John Wiley and Sons Ltd., 1975, Chap. 1

[15] J.A. Stratton, **Electromagnetic Theory**, McGraw-Hill Book Company, 1941, pp.37

[16] M. Morford, R. Lenardon, **Classical Mythology**, Longman, Inc., Second Edition, 1977. See page 265. MEGAERA was one of the underworld furies in Greek mythology.

[17] D.A. Redford, P.F. Cotsworth, Development of Communication Paths Within a Tar Sand Bed, **Shale Oil, Tar Sand and Related Fuels**, 1976, pp. 112

[18] S.M. Farouq Ali, Elements of Thermal Oil Recovery, Distributed as lecture notes for MNL E 568, University of Alberta, Fall 1980.

[19] W.F. Ames, **Numerical Methods of Partial Differential Equations**, Academic Press, 1977

[20] D. Potter, **Computational Physics**, John Wiley and Sons Ltd., 1973

[21] B. Carnahan, H.A. Luther, J.O. Wilkes, **Applied Numerical Methods**, John Wiley and Sons Inc., 1969, Chapter 7

[22] D. Gardiner, Applied Electromagnetics group, Dept. of Electrical Engineering, University of Alberta, Private Communication.

[23] J. Mathews, R.L. Walker, **Mathematical Methods of Physics**, W.A. Benjamin, Inc., 1970

[24] H.A. Harvey, S.A. El-Feky, Selective Heating Could Boost Oil Recovery, **Oil and Gas Journal**, v. 76, no. 46, 13 Nov. 1978.

[25] K.V. Roberts, An Introduction to the Olympus System, **Computer Physics Communications**, v. 7, 1974, pp. 237

[26] J.P. Christiansen, K.V. Roberts, Olympus - A Standard Control and Utility Package for Initial-Value Fortran Programs, **Computer Physics Communications**, v. 7, 1974, pp. 245

Appendix I - Listing of the Program MEGAERA

```
C
C
C
C
C
C
C                             ********************
C                             *                  *
C                             *  M E G A E R A   *
C                             *                  *
C                             ********************
C
C
C                   SIMULATION OF ELECTRICAL CONDUCTION HEATING
C                   -------------------------------------------
C                         OF OILSAND AND OTHER MATERIALS
C                         ------------------------------
C
C
C                             ALLAN HIEBERT, AUGUST 1980
C
C                             WRITTEN FOLLOWING THE OLYMPUS
C                             PROGRAMING SYSTEM DEVELOPED AT
C                             CULHAM LABORITORIES, ENGLAND.
C
C
C
C
C
C
C
C
C-----------------------------------------------------------------------
C                                 INDEX OF SUBPROGRAMS
C
C
C                                 MAIN CONTROL.          CLASS 0
C
C  MAIN      FORTRAN MAIN PROGRAM                                  0.0
C  BASIC     INITIALIZE BASIC CONTROL DATA                         0.1
C  MODIFY    MODIFY BASIC DATA IF REQUIRED                         0.2
C  COTROL    CONTROL THE RUN                                       0.3
C  USER      PRINT USER, TIME AND DATE TO DIARY                    0.4
C
C                                 PROLOGUE.              CLASS 1
C
C  LABRUN    LABEL THE RUN                                         1.1
C  CLEAR     CLEAR VARIABLES AND ARRAYS                            1.2
C  PRESET    SET DEFAULT VALUES                                    1.3
C  DATA      DEFINE DATA SPECIFIC TO RUN                           1.4
C  AUXVAL    SET AUXILIARY VALUES                                  1.5
C  INITAL    DEFINE PHYSICAL INITIAL CONDITIONS                    1.6
C  RESUME    RESUME FROM PREVIOUS RECORD                           1.7
C  START     START OR RESTART THE RUN                              1.8
C  EPARAM    CALCULATE ITERATION PARAMETERS FOR A.D.I.P.           1.9
C  TCOEFF    CALCULATE CONSTANTS FOR THE THERMAL EQUATION          1.10
C
C                                 CALCULATION.           CLASS 2
C
C  STEPON    STEP ON THE CALCULATION                               2.1
```

```
C  ELEPOT    CALCULATE THE ELECTRICAL POTENTIAL                       2.2
C  ELECOF    CALCULATE COEFFICENTS OF DIFFERENCE EQUATION             2.3
C  THOMAS    SOLVE TRIDIAGONAL SYSTEM BY THOMAS ALGORITHM             2.4
C  MAXDIF    FIND THE MAXIMUM DIFFERENCE BETWEEN TEM3, TEM1           2.5
C  ECOND     CALCULATE THE ELECTRICAL CONDUCTIVITY                    2.6
C  QCALC     CALCULATE THE HEATING RATES FOR THE GRID                 2.7
C  TCALC     SOLVE THE HEAT DIFFUSION EQUATION                        2.8
C  ENGBAL    CALCULATE THE ENERGY BALANCE FOR THE TIMESTEP            2.9
C  ELECUR    CALCULATE THE CURRENT THROUGH A SURFACE                  2.10
C
C                                  OUTPUT.                    CLASS 3
C OUTPUT(1) CONTROL THE OUTPUT                                        3.1
C OUTGRD(1) OUTPUT ONE OF THE GLOBAL VARIABLE ARRAYS                  3.2
C ALINTP(7) DO A LINEAR INTERPOLATION                                 3.3
C OUTINT(2) INTERPOLATE AND PRINT A GOBAL VARIABLE                    3.4
C OUTR      OUTPUT A LABEL AND A REAL VARIABLE                        3.5
C OUTI      OUTPUT A LABEL AND AN INTEGER VARIABLE                    3.6
C OUTH      OUTPUT A LABEL AND A HOLLERITH VARIABLE                   3.7
C OUTTAP(1) OUTPUT RUN INFORMATION TO MAGNETIC TAPE                   3.8
C
C
C                                  EPILOGUE.                  CLASS 4
C TESEND    TEST FOR COMPLETION OF RUN                                4.1
C ENDRUN    TERMINATE THE RUN                                         4.2
C
C
C                                  DIAGNOSTICS.               CLASS 5
C REPORT(3) CONTROL THE DIAGNOSTICS                                   5.1
C CLIST(2)  PRINT COMMON VARIABLES                                    5.2
C ARRAYS(2) PRINT COMMON ARRAYS                                       5.3
C
C
C                                  UTILITIES.                 CLASS U
C MESAGE(1) PRINT 48-CHARACTER MESSAGE ON OUTPUT CHANNEL              U.1
C PAGE      FETCH NEW PAGE ON OUTPUT CHANNEL                          U.2
C BLINES(1) INSERT BLANK LINES ON OUTPUT CHANNEL                      U.3
C RVAR(2)   PRINT NAME AND VALUE OF REAL VARIABLE                     U.4
C IVAR(2)   PRINT NAME AND VALUE OF INTEGER VARIABLE                  U.5
C HVAR(2)   PRINT NAME AND VALUE OF HOLLERITH VARIABLE                U.6
C LVAR(2)   PRINT NAME AND VALUE OF LOGICAL VARIABLE                  U.7
C RARRAY(3) PRINT NAME AND VALUES OF REAL ARRAY                       U.8
C IARRAY(3) PRINT NAME AND VALUES OF INTEGER ARRAY                    U.9
C HARRAY(3) PRINT NAME AND VALUES OF HOLLERITH ARRAY                  U.10
C REPTHD(3) PRINT HEADING FOR DIAGNOSTIC REPORT                       U.11
C RUNTIM    UPDATE CPU TIME AND PRINT IT                              U.12
C DAYTIM    PRINT DATE AND TIME                                       U.13
C RESETR(3) RESET REAL ARRAY TO SPECIFIED VALUE                       U.14
C RESETI(3) RESET INTEGER ARRAY TO SPECIFIED VALUE                    U.15
C RESETH(3) RESET HOLLERITH ARRAY TO SPECIFIED VALUE                  U.16
C JOBTIM    FETCH ALLOCATED JOBTIME                                   U.17
C LARRAY(3) PRINT NAME AND VALUES OF LOGICAL ARRAY                    U.18
C RESETL(3) RESET LOGICAL ARRAY TO SPECIFIED VALUE                    U.19
C RARAY2    PRINT DOUBLY-SUBSCRIPTED ARRAY                            U.20
C SCALER(3) SCALE A REAL ARRAY BY A REAL VALUE                        U.21
C SCALEI(3) SCALE AN INTEGER ARRAY BY AN INTEGER VALUE                U.22
C COPYR(5)  COPY ONE REAL MATRIX INTO ANOTHER                         U.23
C COPYI(5)  COPY ONE INTEGER MATRIX INTO ANOTHER                      U.24
C SIGNR(2)  CHANGE THE SIGN OF A REAL MATRIX                          U.25
C SIGNI(2)  CHANGE THE SIGN OF AN INTEGER MATRIX                      U.26
```

```
C  DUMCOM(3) DUMP SELECTED COMMON BLOCKS                                 U.27
C
C
CL                         C1.1.    BASIC SYSTEM PARAMETERS
C      VERSION 2B                14.8.73       KVR/MMH         CULHAM
C       COMMON/COMBAS/
C
C
C  ALTIME       TIME ALLOCATED TO JOB. (MINS)                      R   1.1
C  CPTIME       CPU TIME USED SO FAR ON THIS JOB. (MINS)           R   1.1
C  LABEL1(12) LABEL DESCRIBING THE RUN                             IA  1.1
C  LABEL2(12) LABEL DESCRIBING THE RUN                             IA  1.1
C  LABEL3(12) LABEL DESCRIBING THE RUN                             IA  1.1
C  LABEL4(12) LABEL DESCRIBING THE RUN                             IA  1.1
C  LABEL5(12) LABEL AVAILABLE TO PROGRAMMER                        IA  1.1
C  LABEL6(12) LABEL AVAILABLE TO PROGRAMMER                        IA  1.1
C  LABEL7(12) LABEL RESERVED FOR SYSTEM USE                        IA  1.1
C  LABEL8(12) LABEL RESERVED FOR SYSTEM USE                        IA  1.1
C  NDIARY       CHANNEL FOR DIARY                                  I   1.1
C  NIN          CURRENT INPUT CHANNEL                              I   1.1
C  NLEDGE       CHANNEL FOR RESTART RECORDS                        I   1.1
C  NLEND        .TRUE. IF RUN TO BE TERMINATED                     L   1.1
C  NLRES        .TRUE. IF RUN TO BE RESTARTED                      L   1.1
C  NONLIN       CHANNEL FOR ONLINE INPUT-OUTPUT                    I   1.1
C  NOUT         CURRENT OUTPUT CHANNEL                             I   1.1
C  NPRINT       CHANNEL FOR PRINTED OUTPUT                         I   1.1
C  NPUNCH       CHANNEL FOR CARD OUTPUT (OR EQUIVALENT)            I   1.1
C  NREAD        CHANNEL FOR CARD INPUT                             I   1.1
C  NREC         CURRENT RECORD NUMBER                              I   1.1
C  NRESUM       RESUME FROM RECORD ON THIS CHANNEL                 I   1.1
C  NRUN         MAXIMUM NUMBER OF STEPS                            I   1.1
C  NSTEP        CURRENT STEP NUMBER                                I   1.1
C  STIME        START TIME, (MINS)                                 R   1.1
C
C
C-----------------------------------------------------------------------------
C
C
CL                         C1.9.    DEVELOPMENT AND DIAGNOSTIC PARAMETERS
C      VERSION 1B                14.8.73       KVR/MMH         CULHAM
C       COMMON/COMDDP/
C
C
C
C  MAXDUM       MAXIMUM DIMENSION OF DUMP ARRAYS                   I   1.9
C  MXDUMP       ACTUAL DIMENSION OF DUMP ARRAYS                    I   1.9
C  NADUMP(M)    CODES FOR ARRAY DUMPS                              IA  1.9
C  NCLASS       MOST RECENT CLASS REPORTED                         I   1.9
C  NLCHED       .TRUE. IF CLASS 0 REPORT HEADS REQUIRED            L   1.9
C  NLHEAD(9)    .TRUE. IF CLASSES 1-9 REPORT HEADS REQUIRED        LA  1.9
C  NLOMT1(50) CLASS 1 SUBPROGRAM SELECTOR                          LA  1.9
C  NLOMT2(50) CLASS 2 SUBPROGRAM SELECTOR                          LA  1.9
C  NLOMT3(50) CLASS 3 SUBPROGRAM SELECTOR                          LA  1.9
C  NLREPT       .TRUE. IF ANY REPORT REQUIRED                      L   1.9
C  NPDUMP(M)    CODES FOR DUMPING POINTS                           IA  1.9
C  NPOINT       MOST RECENT POINT REPORTED                         I   1.9
C  NSUB         MOST RECENT SUBPROGRAM REPORTED                    I   1.9
C  NVDUMP(M)    CODES FOR DUMPING VARIABLES                        IA  1.9
C
C-----------------------------------------------------------------------------
```

```
C
C        1.2     /COMGLO/ - GLOBAL VARIABLES
C
C  DELTAX     WIDTHS OF GRID BLOCKS                            R  1.2
C  DELTAY     HEIGHT OF GRID BLOCKS                            R  1.2
C  DTIME      LENGTH OF TIMESTEP ( DELTA TIME)                 R  1.2
C  ELECON     ELECTRICAL CONDUCTIVITY                          R  1.2
C  ELEGEO     ELECTRICAL GEOMETRY                              R  1.2
C                NEGATIVE - DOMAIN POINT
C                ZERO       - NOT IN DOMAIN
C                POSITIVE  - ELECTRODE
C  QTHERM     HEAT GENERATED BY ELECTRIC FIELD                 R  1.2
C  POTENT     ELECTRICAL POTENTIAL                             R  1.2
C  TEMP       TEMPERATURE                                      R  1.2
C  THICK      THICKNESS OF TWO DIMENSIONAL SLICE               R  1.2
C  THMGEO     THERMAL GEOMETRY                                 R  1.2
C  TIME       CURRENT VALUE OF THE TIME                        R  1.2
C  XCOORD     X COORDINATES OF GRIDPOINTS                      R  1.2
C  YCOORD     Y COORDINATES OF GRIDPOINTS                      R  1.2
C              ORIGIN AT LOWER LEFT CORNER OF (1,1) BLOCK
C  XMIN       MINIMUM X  FOR INTERPOLATED PRINTOUT             R  1.2
C  XMAX       MAXIMUM X  FOR INTERPOLATED PRINTOUT             R  1.2
C  YMIN       MINIMUM Y  FOR INTERPOLATED PRINTOUT             R  1.2
C  YMAX       MAXIMUM Y  FOR INTERPOLATED PRINTOUT             R  1.2
C
C-----------------------------------------------------------------------
C
C        1.3      /COMELE/ - ELECTRICAL COEFFICENTS
C
C  ECXM       COEFFICENT OF FINITE DIFFERENCE EQU.             R  1.3
C  ECXP       COEFFICENT IN POSITIVE X DIRECTION               R  1.3
C  ECYM            "      "   NEGATIVE Y     "                 R  1.3
C  ECYP            "      "   POSITIVE Y DIRECTION             R  1.3
C  EXMXP      SUM OF THE TWO X DIRECTION COEFFICENTS           R  1.3
C  EYMYP      SUM OF THE TWO Y DIRECTION COEFFICENTS           R  1.3
C  ERHS       RIGHT HAND SIDE OF DIFFERENCE EQU.               R  1.3
C  ELEVOL     VOLTAGES OF DIFFERENT ELECTRODES                 R  1.3
C  ELEALP     COEFFICENTS FOR CONDUCTIVITY CALC.               R  1.3
C  ELEBET          "       "        "         "                R  1.3
C  ELEPAR     ITERATION PARAMETERS FOR A.D.I.P.                R  1.3
C  EPSELE     CONVERGENCE CRITERIA FOR POTENTIAL               R  1.3
C
C-----------------------------------------------------------------------
C
C        1.4    /COMTHM/ - THERMAL CONDUCTIVITY AND CAPACITY
C
C  THMCON     THERMAL CONDUCTIVITIES                           R  1.4
C  THMCAP     VOLUMETRIC HEAT CAPACITY                         R  1.4
C  THMTEM     VECTOR OF CONSTANT TEMPERATURES                  R  1.4
C  TCYP       COEFFICENT OF THE THERMAL EQU. ( Y POS. )        R  1.4
C  TCXP            "      "   "     "     "   ( X POS. )       R  1.4
C  TCXM            "      "   "     "     "   ( X NEG. )       R  1.4
C  TCYM            "      "   "     "     "   ( Y NEG. )       R  1.4
C  TCX        SUM OF X COEFFICENTS                             R  1.4
C  TCY        SUM OF Y COEFFICENTS                             R  1.4
C  TCRHS      TERMS ON CONSTANT TEMPERATURE BOUNDARY           R  1.4
C
C-----------------------------------------------------------------------
C
C        1.5   /COMTEM/ - TEMPERARY ARRAYS
```

```
C
C  TEM1         TWO DIMENSION ARRAY FOR A.D.I.                  R  1.5
C  TEM2                                                         R  1.5
C  TEM3                                                         R  1.5
C  T1           ONE DIMENSION ARRAY FOR THOMAS ALGOR.           R  1.5
C  T2                                                           R  1.5
C  T3                                                           R  1.5
C  T4                                                           R  1.5
C  T5                                                           R  1.5
C                                                               R  1.5
C-----------------------------------------------------------------------
C
C               1.6  /COMDIM/ - INTEGER COMMON BLOCK
C
C  NX           ACTUAL NUMBER OF GRID BLOCKS IN X DIR.          I  1.6
C  NY           ACTUAL NUMBER OF GRID BLOCKS IN Y DIR.          I  1.6
C  NDX          ACTUAL DIMENSION OF ARRAYS                      I  1.6
C  NDY          ACTUAL DIMENSION OF ARRAYS                      I  1.6
C  NEQU         NUMBER OF EQUATION FOR THOMAS ALGOR.            I  1.6
C  NALPHA       NUMBER OF ITERATION PARAMETERS USED             I  1.6
C  MITER        CURRENT ITERATION COUNTER FOR POTENTIAL         I  1.6
C  MMAX         MAXIMUM ALLOWED VALUE OF MITER                  I  1.6
C  NREG         NUMBER OF DIFFERENT REGIONS IN PROBLEM          I  1.6
C  NGEO         0 IF IN VERTICAL MODE, 1 IN HORIZONTAL          I  1.6
C  NERROR       ERROR RETURN CODE FROM TIMESTEP ROUTINES        I  1.6
C  NTYPE        INDICATES TYPE OF HEATING                       I  1.6
C                  1- CONSTANT VOLTAGE
C                  2- CONSTANT CURRENT
C                  3- CONSTANT POWER
C                  4- NO HEATING
C  NSPO         NUMBER OF TIME STEPS PER OUTPUT                 I  1.6
C  NSTO         NUMBER OF TIMESTEPS BETWEEN TAPE STORES         I  1.6
C  NPX          NUMBER OF X INTERVALS FOR INTERPOLATION         I  1.6
C  NPY          NUMBER OF Y INTERVALS FOR INTERPOLATION         I  1.6
C  NIJ          INDICATES IF SURFACE OF CURRENT INTEGERAL
C               IS PERPENDICULAR TO X OR Y                      I  1.6
C  NCI          INDICE OF GRIDLINE WHICH INTEGRAL IS ALONG      I  1.6
C
C-----------------------------------------------------------------
C
C               1.7   /COMCON/ - HEATING CONTROL VARIABLES
C
C  HTIME        TOTAL TIME OF HEATING                           R  1.7
C  CTIME        TOTOL TIME OF COOLING                           R  1.7
C  CCUR         CONSTANT VALUE OF CURRENT                       R  1.7
C  CPOW         CONSTANT VALUE OF CURRENT                       R  1.7
C  DTEMP        MAXIMUM RELATIVE CHANGE IN TEMPERATURE          R  1.7
C  DELT         ACTUAL CHANGE IN TEMPERATURE FOR TIMESTEP       R  1.7
C  CUR          CURRENT CALCULATED FORM CURRENT INTEGRAL        R  1.7
C  VOLTS        ACTUAL VOLTAGE ACROSS ELECTRODES                R  1.7
C  CVOL         VOLTAGE USED IN PROBLEM DEFINITION              R  1.7
C  RESIST       RESISTANCE BETWEEN ELECTRODES                   R  1.7
C  POWER        ELECTRICAL POWER INPUT DURING TIMESTEP          R  1.7
C  TINIT        INITIAL TEMPERATURE OF THE FORMATION            R  1.7
C  DEENG        ELECTRICAL ENERGY INPUT FOR THE TIMESTEP        R  1.7
C  TEENG        TOTAL ELECTRICAL ENERGY INPUT FOR THE RUN       R  1.7
C  DQENG        HEAT PRODUCED FOR THE TIMESTEP                  R  1.7
C  TQENG        TOTAL HEAT PRODUCED IN THE RUN                  R  1.7
C  QENG         VECTOR OF HEATING IN EACH REGION                R  1.7
C  TTENG        ENERGY IN THE TEMPERATURE CHANGE                R  1.7
```

```
C   TENG        VECTOR OF ENERGY OF TEMP. CHANGE BY REGION  R  1.7
C   SCALE       SCALE FACTOR FOR CONSTANT CURRENT AND POWER R  1.7
C
C----------------------------------------------------------------
C
C
C 0.0  FORTRAN MAIN PROGRAM
C
C
C/ INSERT COMBAS
C
C     TIME ALLOCATED TO JOB
         CALL JOBTIN(ALTIME)
C
C     SET UP THE BASIC CONTROL DATA
         CALL BASIC
C
C     PRINT DATE AND TIME
          CALL PAGE
         CALL DAYTIM
C
C     CONTROL THE RUN
         CALL COTROL
C
         STOP
         END
C
C
C----------------------------------------------------------
C
         SUBROUTINE BASIC
C
C 0.1  INITIALIZE BASIC DATA
C
C
C/ INSERT COMBAS
C/ INSERT COMDOP
      DATA         IBLANK/4H    /
CL                1.          GENERAL OLYMPUS DATA
C
CL                   1.1          BASIC SYSTEM PARAMETERS
C     CPU - TIME USED SO FAR
         CPTIME=0.0
C
C     CLEAR ALL 8 LABEL ARRAYS
         IL = 8 * 12
         CALL RESETH(LABEL1,IL,IBLANK)
C
C     INPUT-OUTPUT CHANNELS
         NLEDGE = 30
         NONLIN = 1
         NPUNCH=7
         NPRINT=6
         NREAD=5
         NDIARY=NPRINT
         NIN=NREAD
         NOUT=NPRINT
C
C     TIMESTEP CONTROL
         NRUN = 1
```

```
        NSTEP=0
C
C     RESTART CONTROL
        NREC = 1
        NRESUM = NLEDGE
C
C     LOGICAL SWITCHES
        NLEND=.FALSE.
        NLRES=.FALSE.
C
CL               1.9      DIAGNOSTIC AND DEVELOPMENT PARAMETERS
        MAXDUM = 20
C     MAXIMUM DIMENSIONS OF DUMP ARRAYS
        MXDUMP = 10
C     RESET DUMP ARRAYS
        CALL RESETI(NADUMP,MAXDUM,0)
        CALL RESETI(NPDUMP,MAXDUM,0)
        CALL RESETI(NVDUMP,MAXDUM,0)
C     TRACER VARIABLES
        NCLASS = 0
        NSUB = 1
        NPOINT = 1
C     LOGICAL SWITCHES
        NLCHED = .FALSE.
        NLREPT = .FALSE.
C     REPORT HEADS FOR CLASSES 1-9
        CALL RESETL(NLHEAD,9,.FALSE.)
C     RESET CLASS 1,2,3 SUBPROGRAM SELECTOR ARRAY
        CALL RESETL(NLOMT1,50,.FALSE.)
        CALL RESETL(NLOMT2,50,.FALSE.)
        CALL RESETL(NLOMT3,50,.FALSE.)
C
C     USER INTERFACE
        CALL MODIFY
C
        RETURN
        END
C
C-----------------------------------------------------------------
C
        SUBROUTINE MODIFY
C
C 0.2  MODIFY BASIC DATA IF REQUIRED
C
C/ INSERT COMBAS
C/ INSERT COMODP
C
      NAMELIST/INMOD/NONLIN,NRUN,NREC,NRESUM,NLRES
C
C     PRINT USER, TIME AND DATE TO DIARY FILE
        CALL USER
C
C     SET DEFAULT VALUE FOR NRESUM
        NRESUM=2
C
C     INPUT NAMELIST TO MODIFY BASIC DATA
        READ(NREAD,INMOD)
        IF(.NOT.NLRES) NREC=1
      RETURN
      END
```

```
C
C------------------------------------------------------------------------
C
      SUBROUTINE COTROL
C
C 0.3  CONTROL THE RUN
C
C     VERSION 2B          17/12/73      KVR/MHH         CULHAM
C
C/ INSERT COMBAS
C/ INSERT COMDDP
CL                1.          PROLOGUE
C
      IF(NLRES) GO TO 170
C
C                 A.          NEW RUN
C
CL                   1.1      LABEL THE RUN
  110 CALL LABRUN
C
CL                   1.2      CLEAR VARIABLES AND ARRAYS
  120 CALL CLEAR
C
CL                   1.3      SET DEFAULT VALUES
  130 CALL PRESET
C
CL                   1.4      DEFINE DATA SPECIFIC TO RUN
  140 CALL DATA
C
CL                   1.5      SET AUXILIARY VALUES
  150 CALL AUXVAL
C
CL                   1.6      DEFINE PHYSICAL INITIAL CONDITIONS
  160 CALL INITAL
      GO TO 180
C
C                 B.          RESUME A PREVIOUS RUN
C
CL                   1.7      PICK UP RECORD, MODIFY REQUIRED PARAMETERS
  170 CONTINUE
C    LABEL THE CONTINUATION RUN
      CALL LABRUN
C    CLEAR VARIABLES AND ARRAYS
      CALL CLEAR
C    PICK UP RECORD AND PRINT DETAILS
      CALL RESUME
C    READ ANY NEW DATA NEEDED
      CALL DATA
C    MODIFY AUXILIARY VARIABLES AS REQUIRED
      CALL AUXVAL
C
C                 C.          PRELIMINARY OPERATIONS
C
CL                   1.8      START OR RESTART THE RUN
  180 CALL START
C    INITIAL OUTPUT
      CALL OUTPUT(1)
C
C
CL                2.          CALCULATION
```

```
C
CL                     2.1      STEP ON THE CALCULATION
  210    CALL STEPON
C
CL               3.         OUTPUT
C
CL                     3.1      PERIODIC PRODUCTION OF OUTPUT
  310    CALL OUTPUT(2)
C
CL               4.         EPILOGUE
C
CL                     4.1      TEST FOR COMPLETION OF RUN
  410    CALL TESEND
         IF(.NOT.NLEND) GO TO 210
C
C      FINAL OUTPUT
         CALL OUTPUT(3)
C
CL                     4.2      TERMINATE THE RUN
  420    CALL ENDRUN
C
         RETURN
         END
C
C----------------------------------------------------------------------
C
         SUBROUTINE USER
C
C  0.4  LOG USER CSID, TIME AND DATE TO DIARY FILE
C
      INTEGER*4 INAME, ITIME(2), IDATE(3)
         CALL GUINFO('SIGNONID',INAME)
         CALL TIME(4,0,ITIME)
         CALL TIME(5,0,IDATE)
         CALL FTNCMD('ASSIGN 18=HADI:ULOG',19)
         WRITE(18,20) INAME,ITIME(1),ITIME(2),(IDATE(J),J=1,3)
 20      FORMAT(' MEGAERA',5X,A4,5X,2A4,5X,3A4)
         RETURN
      END
C
C----------------------------------------------------------------------
C
         SUBROUTINE LABRUN
C
C 1.1  LABEL THE RUN
C
C/ INSERT COMBAS
         NAMELIST/LABELS/LABEL1,LABEL2,LABEL3,LABEL4
C
C      READ IN LABELS
         READ(NREAD,LABELS)
C
C      PRINT PROGRAM HEADING AND LABELS
         CALL BLINES(3)
         WRITE(NPRINT,20)
         WRITE(NPRINT,21)
         WRITE(NPRINT,22)
         CALL BLINES(2)
         CALL MESAGE(LABEL1)
         CALL MESAGE(LABEL2)
```

```
            CALL MESAGE(LABEL3)
            CALL MESAGE(LABEL4)
        RETURN
C
C
 20     FORMAT(35X,'P R O G R A M   M E G A E R A')
 21     FORMAT(35X,'-----------------------------')
 22     FORMAT('0',34X,'ADH',14X,'AUGUST, 1980')
C
        END
C
C---------------------------------------------------------------
C
          SUBROUTINE CLEAR
C
C 1.2  CLEAR VARIABLES AND ARRAYS
C
C
C/ INSERT COMGLO
C/ INSERT COMELE
C/ INSERT COMTHM
C/ INSERT COMTEM
C/ INSERT COMCON
C/ INSERT COMDIM
C
          NDX=50
          NDY=50
          NV=NDX
          IF(NDY.GT.NDX) NV=NDY
          NA=NDX*NDY
C
C     CLEAR GLOBAL VARIABLES
          IL=6 * NA + 4 * NV + 7
                  CALL RESETR(ELECON,IL,0.0)
C
C     CLEAR ELECTRICAL COMMON BLOCKS
          IL= 7 * NA + 3 * 10 + 1 * NV
                  CALL RESETR(ECXP,IL,0.0)
C
C     CLEAR THERMAL COMMON BLOCK
          IL= 3 * 10 + 7 * NA
                  CALL RESETR(THMCON,IL,0.0)
C
C     CLEAR TEMPERARY ARRAYS
          IL= 3 * NA + 5 * NV
                  CALL RESETR(TEM1,IL,0.0)
C
C     CLEAR TIMESTEP CONTROL COMMON BLOCK
          IL= 38
                  CALL RESETR(HTIME,IL,0.0)
C
C     CLEAR INTEGER VARIABLES
          IL= 18
                  CALL RESETI(NX,IL,0)
C
        RETURN
        END
C
C---------------------------------------------------------------
C
```

```
            SUBROUTINE PRESET
C
C 1.3  SET DEFAULT VALUES
C
C/ INSERT COMDIM
C/ INSERT COMELE
C/ INSERT COMCON
C/ INSERT COMGLO
C
C        THE ELECTRICAL AND THERMAL GEOMETRY DEFAULTS TO INSULATORS.
C        THIS HAS BEEN PRESET BY SETTING THE ARRAYS ELEGEO AND
C        THMGEO TO ZERO ( AS DONE IN SUBROUTINE CLEAR ).
C
         NDX=50
         NDY=50
         EPSELE=1.E+6
         SCALE=1.0
         THICK=1.0
         MMAX=200
         NSPO=1
         NERROR=0
         NPX=20
         NPY=20
         NIJ=1
         NCI=0
         NSTO=1000
C
            RETURN
            END
C
C--------------------------------------------------------------------
C
            SUBROUTINE DATA
C
C 1.4  DEFINE DATA SPECIFIC TO RUN
C
C/ INSERT COMGLO
C/ INSERT COMELE
C/ INSERT COMTHM
C/ INSERT COMDIM
C/ INSERT COMBAS
C/ INSERT COMCON
C
        DATA HHOR,HVER/'HORI','VERT'/,HELE,HINS/'ELEC',
     #       'INSU'/,HCTE/'CTEM'/
C
C      NAMELISTS
        NAMELIST /INPUT1/
     #              NX,NY,DELTAX,DELTAY,THICK,GEOMET,NREG
        NAMELIST /REGION/
     #              MINI,MAXI,MINJ,MAXJ,EALPHA,EBETA,THMMK,
     #              THMRC,ETYPE,TTYPE,VOLTS,TEMPER
        NAMELIST /INPUT2/
     #             XMIN,XMAX,YMIN,YMAX,NSPO,NSTO,NPX,NPY
        NAMELIST /INPUT3/
     #             TINIT,DTIME,HTIME,NTYPE,CCUR,CPOW,CTIME,
     #             DTEMP,NIJ,NCI,CVOL,EPSELE,MMAX
C
C      RESUMING AN OLD RUN?
            IF(NLRES) GOTO 10
```

```
C
C     READ IN FIRST NAMELIST
            GEOMET=HVER
            READ(NREAD,INPUT1,END=900)
                NALPHA=MAX(NX,NY)
                CALL BLINES(2)
                CALL OUTR('THICKNESS OF 2D SLICE (THICK)   ',THICK)
                CALL OUTH('GEOMETRY OF PROBLEM             ',GEOMET)
                CALL OUTI('NUMBER OF REGIONS (NREG)        ',NREG)
                IF(NREG.GT.9) GOTO 901
                NGEO=0
                IF(GEOMET.EQ.HHOR) NGEO=1
C
C     READ IN AND OVERLAY DIFFERENT REGIONS
           DO 1 K=1,NREG
                READ(NREAD,REGION,END=900)
                CALL BLINES(2)
                CALL OUTI('REGION NUMBER                   ',K)
C
C     FIND ELECTRICAL AND THERMAL REGION TYPES
                IET=-K
                IF(ETYPE.EQ.HELE) IET=K
                IF(ETYPE.EQ.HINS) IET=0
                ITT=-K
                IF(TTYPE.EQ.HINS) ITT=0
                IF(TTYPE.EQ.HCTE) ITT=K
C
C     OVERLAY ELECTRICAL AND THERMAL GEOMETRIES
                DO 2 I=MINI,MAXI
                     DO 3 J=MINJ,MAXJ
                          ELEGEO(I,J)=IET
                          THMGEO(I,J)=ITT
 3                   CONTINUE
 2              CONTINUE
C
C     STORE ELECTRICAL AND THERMAL PROPERTIES OF THE REGION
                CALL OUTH('ELECTRICAL TYPE                 ',ETYPE)
                IF(IET.GE.0) GOTO 4
                   ELEALP(K)=EALPHA
                   ELEBET(K)=EBETA
                   CALL OUTR('TEMP. DEPENDENCE OF CONDUCTIVITY',
     #                   EALPHA)
                   CALL OUTR('CONDUCTIVITY AT 24 CELSIUS      ',EBETA)
 4              IF(IET.GT.0) ELEVOL(K)=VOLTS
                IF(IET.GT.0)
     #          CALL OUTR('VOLTAGE ( BEFORE SCALING )      ',VOLTS)
                CALL OUTH('THERMAL REGION TYPE             ',TTYPE)
                IF(ITT.GE.0) GOTO 5
                   THMCON(K)=THMK
                   THMCAP(K)=THMRC
                   CALL OUTR('THERMAL CONDUCTIVITY            ',
     #                               THMK)
                   CALL OUTR('THERMAL HEAT CAPACITY           ',
     #                               THMRC)
 5              IF(ITT.GT.0) THMTEM(K)=TEMPER
                IF(ITT.GT.0)
     #          CALL OUTR('CONSTANT TEMPERATURE            ',TEMPER)
C
 1         CONTINUE
           XCOORD(1)=DELTAX(1)/2.
```

```
            YCOORD(1)=DELTAY(1)/2.
               DO 6 I=2,NX
  6                 XCOORD(I)=XCOORD(I-1)+(DELTAX(I-1)+DELTAX(I))/2.
               DO 7 J=2,NY
  7                 YCOORD(J)=YCOORD(J-1)+(DELTAY(J-1)+DELTAY(J))/2.
            XMIN=XCOORD(1)
            XMAX=XCOORD(NX)
            YMIN=YCOORD(1)
            YMAX=YCOORD(NY)
C
C     INPUT OUTPUT CONTROL VARIABLES
            READ(NREAD,INPUT2)
               CALL BLINES(2)
               CALL OUTI('NO. OF STEPS PER PRINTED OUTPUT   ',NSPO)
               CALL OUTI('NO. OF STEPS PER TAPE STORAGE     ',NSTO)
               CALL OUTI('NO. OF INTERPOLATION POINTS IN X  ',NPX)
               CALL OUTI('NO. OF INTERPOLATION POINTS IN Y  ',NPY)
               CALL OUTR('X MINIMUM FOR INTERPOLATION       ',XMIN)
               CALL OUTR('X MAXIMUM FOR INTERPOLATION       ',XMAX)
               CALL OUTR('Y MINIMUM FOR INTERPOLATION       ',YMIN)
               CALL OUTR('Y MAXIMUM FOR INTERPOLATION       ',YMAX)
C
C     INPUT HEATING INFORMATION
 10         READ(NREAD,INPUT3)
               IF(EPSELE.GT.1.E+4) EPSELE=CVOL/1.E+5
               DELT=DTEMP
               IF(NCI.EQ.0) GOTO 902
               CALL BLINES(3)
               CALL OUTR('INITIAL TEMPERATURE IN CELSIUS   ',TINIT)
               CALL OUTR('TOTAL HEATING TIME IN SECONDS    ',HTIME)
               CALL OUTR('INITIAL TIMESTEP SIZE IN SECONDS',DTIME)
               CALL OUTI('GRIDLINE FOR CURRENT INTEGRAL ',NCI)
               IF(NTYPE.EQ.1)
     *         CALL OUTR('CONSTANT VOLTAGE                 ',CVOL)
               IF(NTYPE.EQ.2)
     *         CALL OUTR('CONSTANT CURRENT  ( AMPERES )    ',CCUR)
               IF(NTYPE.EQ.3)
     *         CALL OUTR('CONSTANT POWER  ( WATTS )        ',CPOW)
               CALL OUTR('CHANGE IN TEMPERATURE / TIMESTEP ',DTEMP)
               CALL OUTI('MAXIMUM NO. OF ITERATIONS (MMAX)',MMAX)
               CALL OUTR('ELECTRICAL CONVERGENCE CRITERIA ',EPSELE)
               CALL OUTI('NUMBER OF ITERATION PARAMETERS  ',NALPHA)
         RETURN
C
C     ERROR MESSAGES
 900  CALLMESAGE(48H ***  NAMELIST WAS NOT FOUND DURING DATA INPUT  )
             CALL ENDRUN
             STOP
 901  CALLMESAGE(48H ***  ONLY NINE REGIONS MAY BE SPECIFIED        )
             CALL ENDRUN
             STOP
 902  CALLMESAGE(48H ***  NO SURFACE FOR CURRENT INTEGRAL IS GIVEN  )
             CALL ENDRUN
             STOP
         END
C
C------------------------------------------------------------
C
         SUBROUTINE AUXVAL
C
```

```
C 1.5  SET AUXILIARY VALUES
C
           CALL EPARAM
           CALL TCOEFF
C
         RETURN
         END
C
C------------------------------------------------------------
C
         SUBROUTINE INITAL
C
C 1.6  DEFINE PHYSICAL INITIAL CONDITIONS
C
C/ INSERT COMGLO
C/ INSERT COMDIM
C/ INSERT COMCON
C/ INSERT COMTHM
C
C     SET INTIAL TEMPERATURE
            DO 1 J=1,NY
                 DO 1 I=1,NX
                      TEMP(I,J)=TINIT
                      K=THMGEO(I,J)
                      IF(K.GT.0) TEMP(I,J)=THMTEM(K)
 1          CONTINUE
C
         RETURN
         END
C
C------------------------------------------------------------
C
         SUBROUTINE RESUME
C
C 1.7  RESUME FROM PREVIOUS RECORD
C
C/ INSERT COMBAS
C/ INSERT COMGLO
C/ INSERT COMELE
C/ INSERT COMTHM
C/ INSERT COMCON
C/ INSERT COMDIM
C
C     SET DEFAULT VALUES
           NDX=50
           NDY=50
           SCALE=1.0
           NERROR=0
C
C     INPUT RECORD OF RUN GEOMETRY AND MATERIAL PROPERTY DATA
 1    READ(NRESUM,END=900)ELEGEO,THMGEO,DELTAX,DELTAY,THICK,
     #                  XCOORD,YCOORD,XMIN,XMAX,YMIN,YMAX,ELEVOL,
     #                  ELEALP,ELEBET,EPSELE,THMCON,THMCAP,THMTEM,
     #                  TINIT,NX,NY,NALPHA,MMAX,NREG,NGEO,NSPO,
     #                  NSTO,NPX,NPY,NIJ,NCI
C
C     READ UNTIL SPECIFIED RECORD IS INPUT
 2    READ(NRESUM,END=900)NSTEP,TIME,POTENT,QTHERM,TEMP,TEENG,TENG,
     #                 TQENG,QENG,NR,DTIME
           IF(NR.LT.NREC) GOTO 2
           IF(NR.GT.NREC) GOTO 900
           RETURN
C
C     RECORD NOT FOUND
 900  CALLMESAGE(48H    ***   SPECIFIED RECORD NOT FOUND IN RESUME    )
      CALL ENDRUN
      STOP
         END
C
C------------------------------------------------------------
C
         SUBROUTINE START
C
C 1.8  START OR RESTART THE RUN
C
C
         RETURN
         END
C
C--------------------------------------------------------------
C
           SUBROUTINE EPARAM
C
C  1.9  CALCULATE ITERATION PARAMETERS FOR ELECTRICAL A.D.I.P.
C
C/ INSERT COMELE
C/ INSERT COMDIM
C
      DO 3 K=1,NALPHA
           ELEPAR(K)=(SIN(K*3.141528/(2.*NALPHA)))**2
 3    CONTINUE
C
      RETURN
      END
C
C--------------------------------------------------------------
C
           SUBROUTINE TCOEFF
C
C  1.10  CALCULATE THE COEFFICENTS FOR THE HEAT EQUATION
C
C/ INSERT COMGLO
C/ INSERT COMDIM
C/ INSERT COMTHM
C
C     COVER THE GRID
      DO 1 J=1,NY
           JP=J+1
           JM=J-1
           DY=DELTAY(J)
           DO 2 I=1,NX
                IP=I+1
                IM=I-1
C
C     CHECK IF THE BLOCK IS IN THE THERMAL DOMAIN
                K=THMGEO(I,J)
                IF(K.GE.0) GOTO 2
C
C     FIND THE THERMAL CONDUCTIVITY AND CLEAR VARIABLES
                TC=THMCON(-K)
```

```
              DX=DELTAX(I)
              TCRHS(I,J)=0.
              TCX(I,J)=0.
              TCY(I,J)=0.
C
C    FIND COEFFICENTS IN THE X DIRECTION
C    CHECK SURROUNDING BLOCKS
              KP=0
              IF(I.LT.NX) KP=THMGEO(IP,J)
              KM=0
              IF(I.GT.1)  KM=THMGEO(IM,J)
              TCP=TC
              IF(KP.LT.0) TCP=THMCON(-KP)
              TCM=TC
              IF(KM.LT.0) TCM=THMCON(-KM)
              RP=TC/TCP
              RM=TC/TCM
C
C    DECIDE WHICH CASE THIS GRID BLOCK IS
C    1. NORMAL BLOCK SURROUNDED BY DOMAIN
              IF(KP.GE.0.OR.KM.GE.0) GOTO 3
                H=(DX+DELTAX(IP))/2.
                HM=(DX+DELTAX(IM))/2.
                TCXP(I,J)=(TC+TCP)/(H*(H+HM))
                TCXM(I,J)=(TC+TCM)/(HM*(H+HM))
C    CHECK FOR CONDUCTIVITY DISCONTINUITIES
                IF(KP.NE.K)
    #             TCXP(I,J)=TC/(.25*(H+HM)*(DX+RP*DELTAX(IP)))
                IF(KM.NE.K)
    #             TCXM(I,J)=TC/(.25*(H+HM)*(DX+RM*DELTAX(IM)))
                TCX(I,J)=TCXM(I,J)+TCXP(I,J)
                GOTO 10
C
C    2. CONSTANT TEMPERATURE BOUNDARY IN ONE DIRECTION
 3            IF(KP.EQ.0.OR.KM.EQ.0) GOTO 4
                H=(DX+DELTAX(IP))/2.
                HM=(DX+DELTAX(IM))/2.
                IF(KP.GT.0) H=DX/2.
                IF(KM.GT.0) HM=DX/2.
                TCXP(I,J)=(TC+TCP)/(H*(H+HM))
                TCXM(I,J)=(TC+TCM)/(HM*(H+HM))
                TCX(I,J)=TCXP(I,J)+TCXM(I,J)
                IF(KP.GT.0) TCRHS(I,J)=TCRHS(I,J)+
    #                                  TCXP(I,J)*THMTEM(KP)
                IF(KP.GT.0) TCXP(I,J)=0.
                IF(KM.GT.0) TCRHS(I,J)=TCRHS(I,J)+
    #                                  TCXM(I,J)*THMTEM(KM)
                IF(KM.GT.0) TCXM(I,J)=0.
                GOTO 10
C
C    3. BOUNDARY IN ONE DIRECTION
 4            IF(KP.GE.0.AND.KM.GE.0) GOTO 900
                TCXM(I,J)=0.
                TCXP(I,J)=0.
                IF(KP.LT.0) TCXP(I,J)=
    #                      (TC+TCP)/(.5*((DX+DELTAX(IP))**2))
                IF(KM.LT.0) TCXM(I,J)=
    #                      (TC+TCM)/(.5*((DX+DELTAX(IM))**2))
                TCX(I,J)=TCXM(I,J)+TCXP(I,J)
C
```

```
C     CALCULATE COEFFICENTS IN Y DIRECTION
  10          KP=0
              IF(J.LT.NY) KP=THMGEO(I,JP)
              KM=0
              IF(J.GT.1)  KM=THMGEO(I,JM)
              TCP=TC
              IF(KP.LT.0) TCP=THMCON(-KP)
              TCM=TC
              IF(KM.LT.0) TCM=THMCON(-KM)
              RP=TC/TCP
              RM=TC/TCM
C
C     DECIDE WHICH CASE THIS GRID BLOCK IS
C     1. NORMAL BLOCK SURROUNDED BY DOMAIN
              IF(KP.GE.0.OR.KM.GE.0) GOTO 5
                 H=(DY+DELTAY(JP))/2.
                 HM=(DY+DELTAY(JM))/2.
                 TCYP(I,J)=(TC+TCP)/(H*(H+HM))
                 TCYM(I,J)=(TC+TCM)/(HM*(H+HM))
C     CHECK FOR CONDUCTIVITY DISCONTINUITIES
                 IF(KP.NE.K)
     #             TCYP(I,J)=TC/(.25*(H+HM)*(DY+DELTAY(JP)*RP))
                 IF(KM.NE.K)
     #             TCYM(I,J)=TC/(.25*(H+HM)*(DY+DELTAY(JM)*RM))
                 TCY(I,J)=TCYM(I,J)+TCYP(I,J)
                 GOTO 2
C
C     2. CONSTANT TEMPERATURE BOUNDARY
   5          IF(KP.EQ.0.OR.KM.EQ.0) GOTO 6
                 H=(DY+DELTAY(JP))/2.
                 HM=(DY+DELTAY(JM))/2.
                 IF(KP.GT.0) H=DY/2.
                 IF(KM.GT.0) HM=DY/2.
                 TCYP(I,J)=(TC+TCP)/(H*(H+HM))
                 TCYM(I,J)=(TC+TCM)/(HM*(H+HM))
                 TCY(I,J)=TCYM(I,J)+TCYP(I,J)
                 IF(KP.GT.0) TCRHS(I,J)=TCRHS(I,J)+
     #                                  TCYP(I,J)*THMTEM(KP)
                 IF(KP.GT.0) TCYP(I,J)=0.
                 IF(KM.GT.0) TCRHS(I,J)=TCRHS(I,J)+
     #                                  TCYM(I,J)*THMTEM(KM)
                 IF(KM.GT.0) TCYM(I,J)=0.
                 GOTO 2
C
C     3. BOUNDARY IN ONE DIRECTION
   6          IF(KP.GE.0.AND.KM.GE.0) GOTO 900
                 TCYP(I,J)=0.
                 TCYM(I,J)=0.
                 IF(KP.LT.0) TCYP(I,J)=
     #                       (TC+TCP)/(.5*((DY+DELTAY(JP))**2))
                 IF(KM.LT.0) TCYM(I,J)=
     #                       (TC+TCM)/(.5*((DY+DELTAY(JM))**2))
                 TCY(I,J)=TCYM(I,J)+TCYP(I,J)
C
C
   2          CONTINUE
   1       CONTINUE
        RETURN
C
C       ERROR RETURN
```

```
 900  CALLMESAGE(48H   ***    ERROR RETURN FROM ROUTINE TCOEF          )
        CALL ENDRUN
        RETURN
      END
C
C-----------------------------------------------------------------
C
        SUBROUTINE STEPON
C
C 2.1  STEP ON THE CALCULATION
C
C/ INSERT COMBAS
C/ INSERT COMGLO
C/ INSERT COMCON
C/ INSERT COMDIM
         NSTEP=NSTEP+1
C
C
C    IF END OF HEATING PERIOD, RESET ELECTRICAL VARIABLES
         IF(ABS(TIME-HTIME).GT.1.) GOTO 1
             NTYPE=4
             CALL RESETR(POTENT,NDX*NDY,0.0)
             CALL RESETR(QTHERM,NDX*NDY,0.0)
             VOLTS=0.
             POWER=0.
             CUR  =0.
             RESIST=0.
C
C    IF NO HEATING, SKIP E-FIELD AND Q CALCULATIONS
 1       IF(NTYPE.EQ.4) GOTO 14
C
C    CALCULATE ELECTRICAL CONDUCTIVITY
           CALL ECOND
           IF(NERROR.NE.0) RETURN
C
C    DESCALE POTENTIAL AND FIND NEW POTENTIAL
           S=1./SCALE
           CALL SCALER(POTENT,NDX*NDY,S)
           CALL ELEPOT
           IF(NERROR.NE.0) RETURN
C
C    SCALE THE POTENTIAL IF CONSTANT CURRENT OR POWER
           CALL ELECUR
           RESIST=CVOL/CUR
           GOTO (10,11,12) ,NTYPE
C
C    CONSTANT VOLTAGE
 10              VOLTS=CVOL
                 POWER=VOLTS*CUR
                 GOTO 13
C
C    CONSTANT CURRENT
 11              SCALE=CCUR/CUR
                 VOLTS=CVOL*SCALE
                 CALL SCALER(POTENT,NDX*NDY,SCALE)
                 POWER=CCUR*VOLTS
                 CUR=CCUR
                 GOTO 13
C
C    CONSTANT POWER
```

```
   12                SCALE=SQRT(CPOW/(CUR*CVOL))
                     VOLTS=SCALE*CVOL
                     POWER=CPOW
                     CUR=CUR*SCALE
                     CALL SCALER(POTENT,NDX*NDY,SCALE)
  C
  C     CALCULATE THE HEATING RATES
   13            CALL QCALC
                 IF(NERROR.NE.O) RETURN
  C
  C     FIND THE TIMESTEP SIZE FOR THE NEXT HEATING STEP
   14        DTIME=DTIME*DTEMP/DELT
             IF(NTYPE.LE.3.AND.(DTIME+TIME).GT.(HTIME-1.E-2))
       #         DTIME=HTIME-TIME
             IF(NTYPE.EQ.4.AND.(DTIME+TIME).GT.(HTIME+CTIME))
       #         DTIME=HTIME+CTIME-TIME
  C
  C     FIND THE NEW TEMPERATURE
            CALL TCALC
                IF(NERROR.NE.O) RETURN
  C
  C     CHECK THE ENERGY BALANCE
            CALL ENGBAL
            TIME=TIME+DTIME
          RETURN
          END
  C
  C-----------------------------------------------------------
  C
           SUBROUTINE ELEPOT
  C
  C  2.2  CALCULATE THE ELECTRICAL POTENTIAL FROM THE
  C       FINITE DIFFERENCE EQUATION, USING THE
  C       ALTERNATING DIRECTION IMPLICIT PROCEDURE.
  C
  C/ INSERT COMGLO
  C/ INSERT COMELE
  C/ INSERT COMTEM
  C/ INSERT COMDIM
  C
  C     FIRST, CALCULATE THE COEFFICENTS FOR THIS TIME STEP
            CALL ELECOF
                IF(NERROR.NE.O) RETURN
  C
  C     USE THE POTENTIAL OF THE LAST STEP AS THE FIRST GUESS
            CALL COPYR(POTENT,1,TEM1,1,NDX*NDY)
  C
  C     RESET THE ITERATION COUNTER
            MITER=O
  C
  C  ***  START OF ITERATION LOOP FOR A.D.I.P.  ***
    1       MITER=MITER+1
  C
  C     CHECK IF THE MAXIMUM NUMBER OF ITERATIONS IS REACHED
            IF(MITER.GT.MMAX) GOTO 900
  C
  C     CHOOSE THE ITERATION PARAMETER
            L=MITER-(MITER/NALPHA)*NALPHA
            IF(L.EQ.O) L=NALPHA
            ALPHA=ELEPAR(L)
```

```
C
C     FIRST STAGE, IMPLICIT IN X.
      DO 2 J=1,NY
         JM=J-1
         IF(J.EQ.1) JM=1
         JP=J+1
         IF(J.EQ.NY) JP=NY
         NEQU=0
         DO 3 I=1,NX
C
C     IF THE BLOCK IS AN ELECTRODE, OR NOT IN THE DOMAIN,
C        SKIP TO THE NEXT BLOCK.
            K=ELEGEO(I,J)
            IF(K.GE.0) GOTO 3
C
C     CALCULATE AND ASSIGN TRIDIAGONAL COEFFICENTS
            NEQU=NEQU+1
            ECSUM=EXMXP(I,J)+EYMYP(I,J)
            T1(NEQU)=ECXM(I,J)
            T2(NEQU)=-(EXMXP(I,J)+ALPHA*ECSUM)
            T3(NEQU)=ECXP(I,J)
            T4(NEQU)=-(ECYM(I,J)*TEM1(I,JM)+ECYP(I,J)*TEM1(I,JP))
     #               +(EYMYP(I,J)-ALPHA*ECSUM)*TEM1(I,J)
     #               -ERHS(I,J)
    3    CONTINUE
C
C     SOLVE THE TRIDIAGONAL SYSTEM
         CALL THOMAS
C
C     PLACE RESULTS IN TEM2
         NEQU=0
         DO 5 I=1,NX
            TEM2(I,J)=0.
            K=ELEGEO(I,J)
            IF(K.GE.0) GOTO 5
            NEQU=NEQU+1
            TEM2(I,J)=T5(NEQU)
    5    CONTINUE
C
    2 CONTINUE
C
C     START OF SECOND STAGE
      DO 6 I=1,NX
         IM=I-1
         IF(I.EQ.1) IM=I
         IP=I+1
         IF(I.EQ.NX) IP=I
         NEQU=0
         DO 7 J=1,NY
C
C     IF NOT IN DOMAIN, SKIP TO NEXT POINT
            K=ELEGEO(I,J)
            IF(K.GE.0) GOTO 7
C
C     CALCULATE AND ASSIGN TRIDIAGONAL COEFFICENTS
            NEQU=NEQU+1
            ECSUM=EXMXP(I,J)+EYMYP(I,J)
            T1(NEQU)=ECYM(I,J)
            T2(NEQU)=-(EYMYP(I,J)+ALPHA*ECSUM)
            T3(NEQU)=ECYP(I,J)
```

```
                 T4(NEQU)=-(ECXM(I,J)*TEM2(IM,J)+ECXP(I,J)*TEM2(IP,J))
      #                   +(EXMXP(I,J)-ALPHA*ECSUM)*TEM2(I,J)
      #                   -ERHS(I,J)
  7         CONTINUE
C
C     SOLVE TRIDIAGONAL SYSTEM
            CALL THOMAS
C
C     PLACE RESULTS IN TEM3
            NEQU=0
            DO 9 J=1,NY
                 TEM3(I,J)=0.
                 K=ELEGEO(I,J)
                 IF(K.GT.0) TEM3(I,J)=ELEVOL(K)
                 IF(K.GE.0) GOTO 9
                 NEQU=NEQU+1
                 TEM3(I,J)=T5(NEQU)
  9         CONTINUE
C
  6      CONTINUE
C
C     CHECK FOR CONVERGENCE
            CALL MAXDIF(1,EPS)
                  IF(EPS.LT.EPSELE) GOTO 10
C
C     NO CONVERGENCE, TRANSFERE TEM3 TO TEM1
            CALL COPYR(TEM3,1,TEM1,1,NDX*NDY)
C
C     LOOP BACK TO START NEXT ITERATION
            GOTO 1
C
C  ***  END OF ITERATION LOOP  ***
C
C     TRANSFERE FINAL SOLUTION TO POTENT
 10         CALL COPYR(TEM3,1,POTENT,1,NDX*NDY)
      RETURN
C
C     ERROR MESSAGE - SOLUTION FAILED TO CONVERGE
 900        NERROR=1
            CALL COPYR(TEM3,1,POTENT,1,NDX*NDY)
            RETURN
      END
C
C-------------------------------------------------------------
C
         SUBROUTINE ELECOF
C
C  2.3  CALCULATION OF COEFFICENTS OF ELECTRICAL DIFFERENCE EQU.
C
C/ INSERT COMGLO
C/ INSERT COMELE
C/ INSERT COMDIM
C
C     FOR EACH GRID BLOCK, CALCULATE THE FOUR COEFFICENTS
       DO 1 J=1,NY
            DY=DELTAY(J)
            DO 2 I=1,NX
C     IF BLOCK NOT IN DOMAIN, THEN SKIP TO NEXT BLOCK
                 K=ELEGEO(I,J)
                 IF(K.GE.0) GOTO 2
```

```
C
                 ERHS(I,J)=0.
                 EXMXP(I,J)=0.
                 EYMYP(I,J)=0.
                 DX=DELTAX(I)
                 S=ELECON(I,J)
C
C     CALCULATE COEFFICENTS FOR X DIRECTION
C     CHECK SURROUNDING BLOCKS
               KP=0
               IF(I.LT.NX) KP=ELEGED(I+1,J)
               KM=0
               IF(I.GT.1)  KM=ELEGED(I-1,J)
               SP=S
               IF(KP.LT.0)  SP=ELECON(I+1,J)
               SM=S
               IF(KM.LT.0)  SM=ELECON(I-1,J)
               RP=S/SP
               RM=S/SM
C
C     DECIDE WHICH CASE THIS GRID BLOCK IS
C     1. NORMAL BLOCK SURROUNDED BY DOMAIN
               IF(KP.GE.0.OR.KM.GE.0) GOTO 3
                 H=(DX+DELTAX(I+1))/2.
                 HM=(DX+DELTAX(I-1))/2.
                 ECXP(I,J)=(S+SP)/(H*(H+HM))
                 ECXM(I,J)=(S+SM)/(HM*(H+HM))
C     CHECK FOR CONDUCTIVITY DISCONTINUITIES
                 IF(KP.NE.K)
     #              ECXP(I,J)=S/(.25*(H+HM)*(DX+RP*DELTAX(I+1)))
                 IF(KM.NE.K)
     #              ECXM(I,J)=S/(.25*(H+HM)*(DX+RM*DELTAX(I-1)))
                 EXMXP(I,J)=ECXM(I,J)+ECXP(I,J)
                 GOTO 10
C
C     2. ELECTRODE IN ONE DIRECTION
 3             IF(KP.EQ.0.OR.KM.EQ.0) GOTO 4
                 H=(DX+DELTAX(I+1))/2.
                 HM=(DX+DELTAX(I-1))/2.
                 IF(KP.GT.0) H=DX/2.
                 IF(KM.GT.0) HM=DX/2.
                 ECXP(I,J)=(S+SP)/(H*(H+HM))
                 ECXM(I,J)=(S+SM)/(HM*(H+HM))
                 EXMXP(I,J)=ECXM(I,J)+ECXP(I,J)
                 IF(KP.GT.0) ERHS(I,J)=ERHS(I,J)+
     #                                   ECXP(I,J)*ELEVOL(KP)
                 IF(KP.GT.0) ECXP(I,J)=0.
                 IF(KM.GT.0) ERHS(I,J)=ERHS(I,J)+
     #                                   ECXM(I,J)*ELEVOL(KM)
                 IF(KM.GT.0) ECXM(I,J)=0.
                 GOTO 10
C
C     3. BOUNDARY IN ONE DIRECTION
 4             IF(KP.GE.0.AND.KM.GE.0) GOTO 900
                 ECXM(I,J)=0.
                 ECXP(I,J)=0.
                 IF(KP.LT.0) ECXP(I,J)=
     #                       (S+SP)/(.5*((DX+DELTAX(I+1))**2))
                 IF(KM.LT.0) ECXM(I,J)=
     #                       (S+SM)/(.5*((DX+DELTAX(I-1))**2))
```

```
                EXMXP(I,J)=ECXM(I,J)+ECXP(I,J)
C
C    CALCULATE COEFFICENTS IN THE Y DIRECTION
 10          KP=O
             IF(J.LT.NY) KP=ELEGEO(I,J+1)
             KM=O
             IF(J.GT.1)  KM=ELEGEO(I,J-1)
             SP=S
             IF(KP.LT.O) SP=ELECON(I,J+1)
             SM=S
             IF(KM.LT.O) SM=ELECON(I,J-1)
             RP=S/SP
             RM=S/SM
C
C    DECIDE WHICH CASE THIS GRID BLOCK IS
C    1. NORMAL BLOCK SURROUNDED BY DOMAIN
             IF(KP.GE.O.OR.KM.GE.O) GOTO 5
               H=(DY+DELTAY(J+1))/2.
               HM=(DY+DELTAY(J-1))/2.
               ECYP(I,J)=(S+SP)/(H*(H+HM))
               ECYM(I,J)=(S+SM)/(HM*(H+HM))
C    CHECK FOR CONDUCTIVITY DISCONTINUITIES
               IF(KP.NE.K)
    #             ECYP(I,J)=S/(.25*(H+HM)*(DY+RP*DELTAY(J+1)))
               IF(KM.NE.K)
    #             ECYM(I,J)=S/(.25*(H+HM)*(DY+RM*DELTAY(J-1)))
               EYMYP(I,J)=ECYM(I,J)+ECYP(I,J)
               GOTO 2
C
C    2. ELECTRODE IN ONE DIRECTION
 5           IF(KP.EQ.O.OR.KM.EQ.O) GOTO 6
               H=(DY+DELTAY(J+1))/2.
               HM=(DY+DELTAY(J-1))/2.
               IF(KP.GT.O) H=DY/2.
               IF(KM.GT.O) HM=DY/2.
               ECYP(I,J)=(S+SP)/(H*(H+HM))
               ECYM(I,J)=(S+SM)/(HM*(H+HM))
               EYMYP(I,J)=ECYM(I,J)+ECYP(I,J)
               IF(KP.GT.O) ERHS(I,J)=ERHS(I,J)+
    #                                ECYP(I,J)*ELEVOL(KP)
               IF(KP.GT.O) ECYP(I,J)=O.
               IF(KM.GT.O) ERHS(I,J)=ERHS(I,J)+
    #                                ECYM(I,J)*ELEVOL(KM)
               IF(KM.GT.O) ECYM(I,J)=O.
               GOTO 2
C
C    3. BOUNDARY IN ONE DIRECTION
 6           IF(KP.GE.O.AND.KM.GE.O) GOTO 900
               ECYM(I,J)=O.
               ECYP(I,J)=O.
               IF(KP.LT.O) ECYP(I,J)=
    #                          (S+SP)/(.5*((DY+DELTAY(J+1))**2))
               IF(KM.LT.O) ECYM(I,J)=
    #                          (S+SM)/(.5*((DY+DELTAY(J-1))**2))
               EYMYP(I,J)=ECYM(I,J)+ECYP(I,J)
C
C
 2         CONTINUE
 1       CONTINUE
     RETURN
```

```
C
C     ERROR RETURN
 900    NERROR=4
        RETURN
        END
C
C------------------------------------------------------------------
C
           SUBROUTINE THOMAS
C
C  2.4  SOLVE A TRIDIAGONAL SYSTEM BY THOMAS'S ALGORITHM
C
C/ INSERT COMTEM
C/ INSERT COMDIM
C
          DIMENSION W(50),G(50)
C
           W(1)=T3(1)/T2(1)
           G(1)=T4(1)/T2(1)
C
           DO 1 I=2,NEQU
                IM=I-1
                DENOM=T2(I)-T1(I)*W(IM)
                W(I)=T3(I)/DENOM
                G(I)=(T4(I)-T1(I)*G(IM))/DENOM
 1         CONTINUE
C
           T5(NEQU)=G(NEQU)
           DO 2 I=2,NEQU
                I1=NEQU+1-I
                IP=I1+1
                T5(I1)=G(I1)-W(I1)*T5(IP)
 2         CONTINUE
C
         RETURN
       END
C
C-----------------------------------------------------------
C
           SUBROUTINE MAXDIF(ICODE,EPS)
C
C  2.5  FIND THE MAXIMUM DIFFERENCE BETWEEN TEM3, TEM1
C
C/ INSERT COMTEM
C/ INSERT COMDIM
C/ INSERT COMGLO
C
C
       EPS=0.
       DO 1 J=1,NY
            DO 2 I=1,NX
                  IF(ICODE.EQ.1) K=ELEGEO(I,J)
                  IF(ICODE.EQ.2) K=THMGEO(I,J)
                       IF(K.GE.0) GOTO 2
                  DIF=ABS(TEM3(I,J)-TEM1(I,J))
                  IF(ICODE.EQ.2)
     #            DIF=DIF/ABS(TEM1(I,J))
                  EPS=AMAX1(DIF,EPS)
 2          CONTINUE
 1     CONTINUE
```

```
      RETURN
      END
C
C-----------------------------------------------------------------
C

           SUBROUTINE ECOND
C
C  2.6  CALCULATE THE ELECTRICAL CONDUCTIVITY
C
C     THE ELECTRICAL CONDUCTIVITY IS ASSUMED TO BE TEMPERATURE
C     DEPENDENT, ACCORDING TO THE FORMULA:
C
C              SIGMA(T)=BETA(1+ALPHA(T-24))
C
C     WHERE T IS IN DEGREES CELCIUS.
C
C/ INSERT COMGLO
C/ INSERT COMELE
C/ INSERT COMDIM
C
C
      DO 1 J=1,NY
           DO 2 I=1,NX
                K=ELEGEO(I,J)
                IF(K.GE.O) GOTO 2
C
C     FIND THE COEFFICENTS FOR THE CONDUCTIVITY
                K=-K
                ALPHA=ELEALP(K)
                BETA=ELEBET(K)
                T=TEMP(I,J)
C
                ELECON(I,J)=BETA*(1.+ALPHA*(T-24.))
C
 2         CONTINUE
 1    CONTINUE
      RETURN
      END
C
C-------------------------------------------------------------
C
           SUBROUTINE QCALC
C
C  2.7  CALCULATE THE HEATING RATE AT EACH GRIDPOINT.
C
C/ INSERT COMDIM
C/ INSERT COMGLO
C
C     FOR EACH GRID POINT
      DO 80 J=1,NY
            JP=J+1
            JM=J-1
          DO 81 I=1,NX
                IM=I-1
                IP=I+1
                QTHERM(I,J)=O.
C
C     CHECK IF THIS GRIDBLOCK IS IN THE DOMAIN
                K=ELEGEO(I,J)
                IF(K.GE.O) GOTO 81
```

```
C
C     CHECK BLOCKS IN THE X DIRECTION
                  KP=0
                  IF(I.LT.NX) KP=ELEGEO(IP,J)
                  KM=0
                  IF(I.GT.1)  KM=ELEGEO(IM,J)
C
C     CALCULATION WHEN SURROUNDING BLOCKS ARE DOMAIN BLOCKS
                  IF(KP.EQ.0.OR.KM.EQ.0.OR.(KP.NE.K.AND.KM.EQ.K.AND
     #            .KP.LT.0).OR.(KM.NE.K.AND.KP.EQ.K.AND.KM.LT.0))GOTO 2
                      H=(DELTAX(I)+DELTAX(IP))/2.
                      HM=(DELTAX(I)+DELTAX(IM))/2.
                      IF(KP.GT.0) H=DELTAX(I)/2.
                      IF(KM.GT.0) HM=DELTAX(I)/2.
                      HS=H**2
                      HMS=HM**2
                      EX=(POTENT(IP,J)*HMS+(HS-HMS)*POTENT(I,J)
     #                     -POTENT(IM,J)*HS)/(H*HM*(H+HM))
                  GOTO 3
C
C     SPECIAL CASES
 2                IF(KP.GE.0.AND.KM.GE.0) GOTO 900
                  IF((KP.EQ.0.AND.KM.LT.0).OR.KM.EQ.K)
     #    EX=(2./(DELTAX(I)+DELTAX(IM)))*(POTENT(I,J)-POTENT(IM,J))
                  IF((KP.LT.0.AND.KM.EQ.0).OR.KP.EQ.K)
     #    EX=(2./(DELTAX(I)+DELTAX(IP)))*(POTENT(IP,J)-POTENT(I,J))
C
 3                CONTINUE
C
C     CHECK BLOCKS IN THE Y DIRECTION
                  KP=0
                  IF(J.LT.NY) KP=ELEGEO(I,JP)
                  KM=0
                  IF(J.GT.1)  KM=ELEGEO(I,JM)
C
C     CALCULATION WHEN SURROUNDED BY DOMAIN BLOCKS
                  IF(KP.EQ.0.OR.KM.EQ.0.OR.(KP.NE.K.AND.KM.EQ.K.AND
     #            .KP.LT.0).OR.(KM.NE.K.AND.KP.EQ.K.AND.KM.LT.0))GOTO 4
                      H=(DELTAY(J)+DELTAY(JP))/2.
                      HM=(DELTAY(J)+DELTAY(JM))/2.
                      IF(KP.GT.0) H=DELTAY(J)/2.
                      IF(KM.GT.0) HM=DELTAY(J)/2.
                      HS=H**2
                      HMS=HM**2
                      EY=(POTENT(I,JP)*HMS+(HS-HMS)*POTENT(I,J)
     #                     -POTENT(I,JM)*HS)/(H*HM*(H+HM))
                  GOTO 5
C
C     SPECIAL CASES
 4                IF(KP.GE.0.AND.KM.GE.0) GOTO 900
                  IF((KP.EQ.0.AND.KM.LT.0).OR.KM.EQ.K)
     #    EY=(2./(DELTAY(J)+DELTAY(JM)))*(POTENT(I,J)-POTENT(I,JM))
                  IF((KP.LT.0.AND.KM.EQ.0).OR.KP.EQ.K)
     #    EY=(2./(DELTAY(J)+DELTAY(JP)))*(POTENT(I,JP)-POTENT(I,J))
C
 5                CONTINUE
C
C     CALCULATE HEATING
                  QTHERM(I,J)=ELECON(I,J)*(EX**2+EY**2)
C
```

```
81            CONTINUE
 80       CONTINUE
       RETURN
C
C     ERROR RETURN
900         NERROR=2
            RETURN
       END
C
C-----------------------------------------------------------
C
.          SUBROUTINE TCALC
C
C  2.8  SOLVE THE TEMPERATURE EQUATION
C
C/ INSERT COMGLO
C/ INSERT COMTHM
C/ INSERT COMTEM
C/ INSERT COMCON
C/ INSERT COMDIM
C
C     COPY THE TEMPERATURE INTO TEM1
           CALL COPYR(TEMP,1,TEM1,1,NDX*NDY)
C
           M=0
 8         M=M+1
C
C     START OF FIRST STAGE
      DO 1 J=1,NY
           JM=J-1
           IF(J.EQ.1) JM=1
           JP=J+1
           IF(J.EQ.NY) JP=NY
           NEQU=0
           DO 2 I=1,NX
C
C     IF BLOCK IS NOT IN DOMAIN, SKIP TO NEXT BLOCK
                K=THMGEO(I,J)
                IF(K.GE.0) GOTO 2
                F1=THMCAP(-K)/(DTIME/2.)
C
C     ASSIGN TRIDIAGONAL COEFFICENTS
                NEQU=NEQU+1
                T1(NEQU)=TCXM(I,J)
                T2(NEQU)=-(TCX(I,J)+F1)
                T3(NEQU)=TCXP(I,J)
                T4(NEQU)=-(TCYM(I,J)*TEM1(I,JM)+TCYP(I,J)*TEM1(I,JP))
     #                   +(TCY(I,J)-F1)*TEM1(I,J)-TCRHS(I,J)
     #                   -QTHERM(I,J)
 2         CONTINUE
C
C     SOLVE TRIDIAGONAL SYSTEM
           CALL THOMAS
C
C     PLACE RESULTS IN TEM2
           NEQU=0
           DO 3 I=1,NX
                TEM2(I,J)=0.
                K=THMGEO(I,J)
                IF(K.GE.0) GOTO 3
                NEQU=NEQU+1
                TEM2(I,J)=T5(NEQU)
 3          CONTINUE
C
 1     CONTINUE
C
C     START OF SECOND STAGE
      DO 4 I=1,NX
           IP=I+1
           IF(I.EQ.NX) IP=NX
           IM=I-1
           IF(I.EQ.1)  IM=I
           NEQU=0
           DO 5 J=1,NY
C
C     IF BLOCK NOT IN DOMAIN, SKIP TO NEXT BLOCK
                K=THMGEO(I,J)
                IF(K.GE.0) GOTO 5
C
C     ASSIGN TRIDIAGONAL COEFFICENTS
                F1=THMCAP(-K)/(DTIME/2.)
                NEQU=NEQU+1
                T1(NEQU)=TCYM(I,J)
                T2(NEQU)=-(TCY(I,J)+F1)
                T3(NEQU)=TCYP(I,J)
                T4(NEQU)=-(TCXM(I,J)*TEM2(IM,J)+TCXP(I,J)*TEM2(IP,J))
     #                   +(TCX(I,J)-F1)*TEM2(I,J)-TCRHS(I,J)
     #                   -QTHERM(I,J)
 5          CONTINUE
C
C     SOLVE THE TRIDIAGONAL SYSTEM
           CALL THOMAS
C
C     PLACE THE RESULTS IN TEM3
           NEQU=0
           DO 6 J=1,NY
                TEM3(I,J)=0.
                K=THMGEO(I,J)
                IF(K.GT.0) TEM3(I,J)=THMTEM(K)
                IF(K.GE.0) GOTO 6
                NEQU=NEQU+1
                TEM3(I,J)=T5(NEQU)
 6          CONTINUE
C
 4    CONTINUE
C
C     FIND THE MAXIMUM RELATIVE CHANGE IN TEMPERATURE
           CALL MAXDIF(2,DELT)
C
C     IF CHANGE LESS THEN TOLERANCE EXIT OUT
           IF(DELT.LE.DTEMP) GOTO 7
C
C     CHANGE TOO BIG, CHANGE DTIME
           IF(M.GT.5) GOTO 900
           DTIME=DTIME*DTEMP/DELT
           DTIME=DTIME-DTIME*.01
           GOTO 8
C
C     COPY NEW TEMPERATURE TO TEMP
 7         CALL COPYR(TEM3,1,TEMP,1,NDX*NDY)
```

```
C
      RETURN
C
C     ERROR RETURN
 900    NERROR=3
        RETURN
      END
C
C---------------------------------------------------------------
C
           SUBROUTINE ENGBAL
C
C  2.9  CALCULATE THE ENERGY BALANCE FOR THE TIMESTEP
C
C     THREE METHODS ARE USED TO CALCULATE THE ENERGY INPUT
C           1.  ELECTRICAL POWER * TIME
C           2.  HEATING RATE * VOLUME * TIME - HEAT FLOW OUT
C           3.  TEMPERATURE RISE * VOLUME * HEAT CAPACITY
C
C     THE LAST TWO METHODS ARE CALCULATED SEPERATELY FOR EACH REGION.
C
C/ INSERT COMGLO
C/ INSERT COMTHM
C/ INSERT COMCON
C/ INSERT COMDIM
C
C     CALCULATE ELECTRICAL ENERGY
           DEENG=POWER * DTIME
           TEENG=TEENG + DEENG
C
C     CALCULATE HEATING ENERGY AND TEMPERATURE RISE ENERGY
           DQENG=0.
           TTENG=0.
           DO 111 K=1,10
 111              TENG(K)=0.
      DO 1 J=1,NY
           DY=DELTAY(J)
           DO 2 I=1,NX
                V=DY*DELTAX(I)*THICK
C
C     CHECK IF BLOCK IN TEMPERATURE DOMAIN
                K=THMGEO(I,J)
                IF(K.GE.0) GOTO 2
C
C     CALCULATE HEAT PRODUCED MINUS HEAT FLOW OUT TO CONSTANT
C     TEMPERATURE BOUNDARIES.
                DQ=QTHERM(I,J)*V*DTIME
                IF(I.EQ.NX) GOTO 3
                  KP=THMGEO(I+1,J)
                  IF(KP.LE.0) GOTO 3
                    DQ1=(TEMP(I,J)-THMTEM(KP))*THMCON(K)*DTIME*
     #                   2.*DY*THICK/DELTAX(I)
                    DQ=DQ-DQ1
 3              IF(I.EQ.1) GOTO 4
                  KM=THMGEO(I-1,J)
                  IF(KM.LE.0) GOTO 4
                    DQ1=(TEMP(I,J)-THMTEM(KM))*THMCON(K)*DTIME*
     #                   2.*DY*THICK/DELTAX(I)
                    DQ=DQ-DQ1
 4              IF(J.EQ.NY) GOTO 5
```

```
                  KP=THMGEO(I,J+1)
                  IF(KP.LE.0) GOTO 5
                    DQ1=(TEMP(I,J)-THMTEM(KP))*THMCON(K)*DTIME*
     #                  2.*DELTAX(I)*THICK/DY
                    DQ=DQ-DQ1
 5              IF(J.EQ.1) GOTO 6
                  KM=THMGEO(I,J-1)
                  IF(KM.LE.0) GOTO 6
                    DQ1=(TEMP(I,J)-THMTEM(KM))*THMCON(K)*DTIME*
     #                  2.*DELTAX(I)*THICK/DY
                    DQ=DQ-DQ1
 6              CONTINUE
                DQENG=DQENG+DQ
                QENG(-K)=QENG(-K)+DQ
C
C     CALCULATE ENERGY INDICATED BY TEMPERATURE RISE.
                DT=(TEMP(I,J)-TINIT)*V*THMCAP(-K)
                TTENG=TTENG+DT
                TENG(-K)=TENG(-K)+DT
C
 2        CONTINUE
 1    CONTINUE
          TQENG=TQENG+DQENG
      RETURN
      END
C
C--------------------------------------------------------
C
          SUBROUTINE ELECUR
C
C  2.10  FIND THE CURRENT THROUGH A SURFACE PERPENDICULAR TO
C        THE X OR Y DIRECTION
C
C     IF NIJ=1, THE SURFACE IS PERPENDICULAR TO X, ALONG THE
C                 GRIDLINE I=NCI.
C     OTHERWISE, THE INTEGRAL IS PERPENDICULAR TO Y, ALONG THE
C                GRIDLINE J=NCI.
C
C/ INSERT COMGLO
C/ INSERT COMDIM
C/ INSERT COMCON
C
C    CHECK ORIENTATION OF INTEGRAL.
          IF(NIJ.NE.1) GOTO 10
C
C    PERPENDICULAR TO X
          H=(DELTAX(NCI)+DELTAX(NCI+1))/2.
          HM=(DELTAX(NCI)+DELTAX(NCI-1))/2.
          HS=H**2
          HMS=HM**2
          CUR=0.
          DO 1 J=1,NY
               S=ELECON(NCI,J)
               E=(HMS*POTENT(NCI+1,J) + (HS-HMS)*POTENT(NCI,J)
     #            - HS*POTENT(NCI-1,J))/(H*HM*(H+HM))
               CUR=CUR+E*S*DELTAY(J)
1         CONTINUE
          CUR=ABS(CUR*THICK)
          RETURN
C
```

```
C     PERPENDICULAR TO Y
   10      H=(DELTAY(NCI)+DELTAY(NCI+1))/2.
           HM=(DELTAY(NCI)+DELTAY(NCI-1))/2.
           HS=H**2
           HMS=HM**2
           CUR=0.
           DO 2 I=1,NX
                S=ELECON(I,NCI)
                E=( HMS*POTENT(I,NCI+1) + (HS-HMS)*POTENT(I,NCI)
     #               - HS*POTENT(I,NCI-1))/(H*HM*(H+HM))
                CUR=CUR+E*S*DELTAX(I)
    2      CONTINUE
           CUR=ABS(CUR*THICK)
           RETURN
      END
C
C-----------------------------------------------------------------------
          SUBROUTINE OUTPUT(ICODE)
C
C 3.1  CONTROL THE OUTPUT
C
C/ INSERT COMBAS
C/ INSERT COMDIM
C/ INSERT COMGLO
C/ INSERT COMCON
C
C     BRANCH TO INITIAL, MIDDLE, AND FINAL OUPUT ROUTINES
          GOTO (1,2,3),ICODE
C
C     INITIAL OUTPUT
 1              IF(NLRES) RETURN
                CALL PAGE
C     OUTPUT ELECTRICAL GRID STRUCTURE
                CALL OUTGRD(1)
                CALL PAGE
C     OUTPUT THERMAL GRID STRUCTURE
                CALL OUTGRD(2)
C     IF TAPE STORAGE IS REQUESTED, STORE GRID ETC. ON TAPE
                IF(NSTO.LT.200) CALL OUTTAP(1)
          RETURN
C
C
C     PERIODIC OUTPUT
C     CHECK IF PRINTED OUTPUT THIS  TIMESTEP
 2        IF(MOD(NSTEP,NSPO).NE.0.AND.NSTEP.NE.1) GOTO 5
 4          CALL PAGE
            CALL OUTI('TIMESTEP NUMBER                          ',NSTEP)
            IF(NSTO.LE.200)
     #      CALL OUTI('CURRENT RECORD NUMBER                    ',NREC)
            CALL BLINES(1)
            CALL OUTR('TIME AT END OF STEP                      ',TIME)
            CALL OUTR('SIZE OF TIMESTEP                         ',DTIME)
            CALL BLINES(1)
            IF(NTYPE.EQ.4) GOTO 6
              CALL OUTI('NO. OF ITERATIONS OF A.D.I.P.   ',MITER)
              CALL OUTR('VOLTAGE BETWEEN ELECTRODES      ',VOLTS)
              CALL OUTR('POWER INPUT TO DOMAIN           ',POWER)
              CALL OUTR('CURRENT BETWEEN ELECTRODES      ',CUR)
              CALL OUTR('RESISTANCE BETWEEN ELECTRODES   ',RESIST)
              CALL BLINES(1)
```

```
  6          WRITE(NPRINT,20)
             WRITE(NPRINT,22) DEENG,DQENG
             WRITE(NPRINT,21) TEENG,TQENG,TTENG
             DO 100 K=1,NREG
                      WRITE(NPRINT,23) K,QENG(K),TENG(K)
  100        CONTINUE
             CALL BLINES(1)
             IF(NTYPE.EQ.4) GOTO 7
                  CALL OUTINT(4,1)
                  IF((25+NPY).GT.45) CALL PAGE
                  CALL OUTINT(5,1)
                  IF((25+2*NPY).GT.50) CALL PAGE
  7          CALL OUTINT(3,1)
C
C      PERIODIC TAPE OUTPUT
  5          IF(NSTO.LE.200.AND.(MOD(NSTEP,NSTO).EQ.0.OR.ICODE.EQ.3))
     #           CALL OUTTAP(2)
       RETURN
C
C
  20    FORMAT(4X,'ENERGY BALANCE      ( JOULES )')
  21    FORMAT(4X,'CUMULATIVE:  ELECTRICAL ENERGY =',1PE12.4,
     #         5X,'HEAT GENERATED =',1PE12.4,5X,
     #         'TEMP. RISE ENERGY =',1PE12.4)
  22  FORMAT(4X,'TIMESTEP   :  ELECTRICAL ENERGY =',1PE12.4,5X,
     #       'HEAT PRODUCED =',1PE12.4)
  23  FORMAT(8X,'REGION =',I3,5X,'HEAT GENERATED =',1PE12.4,5X,
     #           'TEMP. RISE ENERGY =',1PE12.4)
C
C     FINAL OUTPUT
  3          IF(MOD(NSTEP,NSPO).NE.0.AND.NSTEP.NE.1) GOTO 4
             IF(MOD(NSTEP,NSTO).NE.0) GOTO 5
          RETURN
         END
C
C--------------------------------------------------------------
C
          SUBROUTINE OUTGRD(ICODE)
C
C  3.2   OUTPUT ONE OF THE GLOBAL VARIABLE ARRAYS
C
C     ICODE INDICATES WHICH ARRAY IS TO BE OUTPUT
C           1 - ELEGEO, THE ELECTRICAL GEOMETRY
C           2 - THMGEO, THE THERMAL GEOMETRY
C           3 - TEMP  , THE TEMPERATURE
C           4 - POTENT, THE ELECTRICAL POTENTIAL
C           5 - QTHERM, THE HEATING
C           6 - ELECON, THE ELECTRICAL CONDUCTIVITY
C
C/ INSERT COMBAS
C/ INSERT COMDIM
C/ INSERT COMGLO
C
C    PRINT OUT HEADING
         IF(ICODE.EQ.1) WRITE(NPRINT,20)
         IF(ICODE.EQ.2) WRITE(NPRINT,21)
         IF(ICODE.EQ.3) WRITE(NPRINT,22)
         IF(ICODE.EQ.4) WRITE(NPRINT,23)
         IF(ICODE.EQ.5) WRITE(NPRINT,24)
         IF(ICODE.EQ.6) WRITE(NPRINT,25)
         IF(ICODE.LT.3) GOTO 4
C
C   FORMAT STATEMENTS
 20     FORMAT('0',10X,'ELECTRICAL REGIONS OF THE GRID (NOT TO SCALE)')
 21     FORMAT('0',10X,'THERMAL REGIONS OF THE GRID  (NOT TO SCALE)')
 22     FORMAT('0',10X,'TEMPERATURE IN DEGREES CELSIUS')
 23     FORMAT('0',10X,'ELECTRICAL POTENTIAL IN VOLTS')
 24     FORMAT('0',10X,'HEATING RATE IN JOULES/SECOND-METER**3')
 25     FORMAT('0',10X,'ELECTRICAL CONDUCTIVITY IN MHOS/METER')
 26     FORMAT('0     COLUMN ',10(8X,I2))
 27     FORMAT('        X   ',10(3X,F7.3))
 28     FORMAT(' ROW    Y   ')
 29     FORMAT(' ',I3,1X,F7.3,10(3X,F7.2))
 30     FORMAT(' ',I3,1X,F7.3,10(3X,F7.0))
 31     FORMAT(' ',I3,1X,F7.3,10(3X,F7.4))
 32     FORMAT('0')
 33     FORMAT(' ',I2,5X,50(I2))
 34     FORMAT('0COLUMN ',50(I2))
 35     FORMAT('0ROW')
C
         DO 1 I1=1,NX,10
            IT=MIN0(I1+9,NX)
               DO 2 J2=1,NY,50
                  JI=MIN0(J2+49,NY)
C    PRINT PAGE HEADING
            WRITE(NPRINT,26) (I,I=I1,IT)
            WRITE(NPRINT,27) (XCOORD(I),I=I1,IT)
            WRITE(NPRINT,28)
C    PRINT EACH ROW
                DO 3 JM=J2,JI
                   J=NY+1-JM
                     IF(ICODE.EQ.3)
   #       WRITE(NPRINT,29) J,YCOORD(J),(TEMP(I,J),I=I1,IT)
                     IF(ICODE.EQ.4)
   #       WRITE(NPRINT,29) J,YCOORD(J),(POTENT(I,J),I=I1,IT)
                     IF(ICODE.EQ.5)
   #       WRITE(NPRINT,30) J,YCOORD(J),(QTHERM(I,J),I=I1,IT)
                     IF(ICODE.EQ.6)
   #       WRITE(NPRINT,31) J,YCOORD(J),(ELECON(I,J),I=I1,IT)
 3              CONTINUE
                WRITE(NPRINT,32)
 2            CONTINUE
 1          CONTINUE
       RETURN
C
C    OUTPUT THE GEOMETRY DEFINING ARRAYS
 4         WRITE(NPRINT,35)
        DO 5 J=1,NY
           JM=NY+1-J
           IF(ICODE.EQ.1)
   #          WRITE(NPRINT,33) JM,(ELEGEO(I,JM),I=1,NX)
           IF(ICODE.EQ.2)
   #          WRITE(NPRINT,33) JM,(THMGEO(I,JM),I=1,NX)
 5      CONTINUE
        WRITE(NPRINT,34) (I,I=1,NX)
     RETURN
     END
C
C-------------------------------------------------------------------
C
```

```
      FUNCTION ALINTP(X,Y,Z,Z1,Z2,M,N)
C
C  3.3  INTERPOLATE VALUES FROM GRID
C
C     THIS FUNCTION INTERPOLATES A VALUE AT THE POINT (X,Y) FROM
C     THE MATRIX Z. THE INTERPOLATION IS LINEAR.
C
      DIMENSION Z(50,50),Z1(50),Z2(50)
C
C     FIND FIRST X CO-ORDINATE NOT LESS THAN X
C
         DO 1 II=2,M
      I=II-1
      IF(X-Z1(II))3,1,1
C     ============
    1 CONTINUE
    3    XL=X-Z1(I)
         XR=Z1(I+1)-X
         XBASE=XR+XL
C
C     FIND FIRST Y CO-ORDINATE NOT LESS THAN Y
C
         DO 2 JJ=2,N
      J=JJ-1
      IF(Y-Z2(JJ))4,2,2
C     ============
    2 CONTINUE
    4    YL=Y-Z2(J)
         YR=Z2(J+1)-Y
         YBASE=YR+YL
C
C     BILINEAR INTERPOLATION
C
         ALINTP=(YL*(XR*Z(I,J+1)+XL*Z(I+1,J+1))
     #   +YR*(XR*Z(I,J)+XL*Z(I+1,J)))/(XBASE*YBASE)
      RETURN
      END
C
C--------------------------------------------------------------
C
           SUBROUTINE OUTINT(ICODE,IP)
C
C  3.4  INTERPOLATE ONE OF THE GLOBAL VARIABLES
C
C     ICODE - DETERMINES WHICH VARIABLE IS INTERPOLATED ( SEE OUTGRD )
C     IP    = 1, IF PRINTING DESIRED.
C             RESULTS OF ITERPOLATION STORED IN TEM3
C
C/ INSERT COMGLO
C/ INSERT COMTEM
C/ INSERT COMDIM
C/ INSERT COMBAS
C/ INSERT COMELE
C/ INSERT COMTHM
C
C
           TX=XMAX-XMIN
           TY=YMAX-YMIN
           SX=TX/(NPX-1.)
           SY=TY/(NPY-1.)
```

```
C
C     COPY APPROPRIATE ARRAY TO TEM1
            IF(ICODE.EQ.3) CALL COPYR(TEMP,1,TEM1,1,NDX*NDY)
            IF(ICODE.EQ.4) CALL COPYR(POTENT,1,TEM1,1,NDX*NDY)
            IF(ICODE.EQ.5) CALL COPYR(QTHERM,1,TEM1,1,NDX*NDY)
            IF(ICODE.EQ.6) CALL COPYR(ELECON,1,TEM1,1,NDX*NDY)
C
C     DO INTERPOLATION
C     RESULTS STORED IN TEM3. X COORDINATES IN T1. Y COORDINATES IN T2
      DO 1 J=1,NPY
            Y=YMIN+(J-1)*SY
            T2(J)=Y
            DO 2 I=1,NPX
                  X=XMIN+(I-1)*SX
                  T1(I)=X
                  TEM3(I,J)=ALINTP(X,Y,TEM1,XCOORD,YCOORD,NX,NY)
    2       CONTINUE
    1 CONTINUE
C
C     IF NO PRINTING IS DESIRED, RETURN
            IF(IP.NE.1) RETURN
C
C     PRINT OUT ARRAY
C     PRINT OUT HEADING
          IF(ICODE.EQ.3) WRITE(NPRINT,22)
          IF(ICODE.EQ.4) WRITE(NPRINT,23)
          IF(ICODE.EQ.5) WRITE(NPRINT,24)
          IF(ICODE.EQ.6) WRITE(NPRINT,25)
C
C     FORMAT STATEMENTS
   22     FORMAT('0',10X,'TEMPERATURE IN DEGREES CELSIUS')
   23     FORMAT('0',10X,'ELECTRICAL POTENTIAL IN VOLTS')
   24     FORMAT('0',10X,'HEATING RATE IN JOULES/SECOND-METER**3')
   25     FORMAT('0',10X,'ELECTRICAL CONDUCTIVITY IN MHOS/METER')
   26     FORMAT('0    COLUMN ',10(8X,I2))
   27     FORMAT('0          X (M)',10(3X,F7.3))
   28     FORMAT('    Y (M)')
   29     FORMAT(' ',F7.3,5X,10(2X,F8.1))
C
             DO 6 I1=1,NPX,10
                   IT=MIN0(I1+9,NPX)
                         DO 7 J2=1,NPY,50
                               JI=MIN0(J2+49,NPY)
C     PRINT PAGE HEADING
                  WRITE(NPRINT,27) (T1(I),I=I1,IT)
                  WRITE(NPRINT,28)
C     PRINT EACH ROW
                             DO 8 JM=J2,JI
                                   J=NPY+1-JM
                WRITE(NPRINT,29) T2(J),(TEM3(I,J),I=I1,IT)
    8                        CONTINUE
    7                  CONTINUE
    6             CONTINUE
          RETURN
          END
C
C--------------------------------------------------------------
C
          SUBROUTINE OUTR(A,B)
C
```

```
C  3.5  OUTPUT A REAL VARIABLE WITH A LABEL
C
C/ INSERT COMBAS
           REAL A(8),B
           WRITE(NPRINT,20) (A(I),I=1,8),B
 20     FORMAT(4X,8A4,' =',1PE12.4)
           RETURN
       END
C
C---------------------------------------------------------------
C
           SUBROUTINE OUTI(A,B)
C
C  3.6  OUTPUT A LABEL AND AN INTEGER VARIABLE
C
C/ INSERT COMBAS
           REAL A(8)
           INTEGER B
           WRITE(NPRINT,20) (A(I),I=1,8),B
 20    FORMAT(4X,8A4,' =',I12)
           RETURN
       END
C
C---------------------------------------------------------------
C
          SUBROUTINE OUTH(A,B)
C
C  3.7  OUTPUT A LABEL AND A HOLLERITH VARABLE.
C
C/ INSERT COMBAS
           REAL A(8)
           REAL B
           WRITE(NPRINT,20) (A(I),I=1,8),B
 20     FORMAT(4X,8A4,' =',8X,A4)
           RETURN
           END
C
C---------------------------------------------------------------
C
           SUBROUTINE OUTTAP(ICODE)
C
C  3.8  OUTPUT RUN INFORMATION TO MAGNETIC TAPE
C
C/ INSERT COMBAS
C/ INSERT COMGLO
C/ INSERT COMELE
C/ INSERT COMTHM
C/ INSERT COMCON
C/ INSERT COMDIM
C
          GOTO (1,2) , ICODE
C
C     INITIAL STORAGE
 1        WRITE(NONLIN) ELEGEO,THMGEO,DELTAX,DELTAY,THICK,
     #                  XCOORD,YCOORD,XMIN,XMAX,YMIN,YMAX,ELEVOL,
     #                  ELEALP,ELEBET,EPSELE,THMCON,THMCAP,THMTEM,
     #                  TINIT,NX,NY,NALPHA,MMAX,NREG,NGEO,NSPO,
     #                  NSTO,NPX,NPY,NIJ,NCI
          NREC=NREC+1
          RETURN
```

```
C
C     PERIODIC STORAGE
 2        WRITE(NONLIN) NSTEP,TIME,POTENT,QTHERM,TEMP,TEENG,TENG,
     #                  TQENG,QENG,NREC,DTIME
          NREC=NREC+1
          RETURN
      END
C
C---------------------------------------------------------------
C
           SUBROUTINE TESEND
C
C 4.1  TEST FOR COMPLETION OF RUN
C
C/ INSERT COMBAS
C/ INSERT COMGLD
C/ INSERT COMCON
C/ INSERT COMDIM
C
            IF(NERROR.NE.0) CALL ENDRUN
            IF(NSTEP.GE.NRUN) NLEND = .TRUE.
            IF(TIME.GE.(HTIME+CTIME)) NLEND = .TRUE.
         RETURN
         END
C
C------------------------------------------------------------------
C
           SUBROUTINE ENDRUN
C
C 4.2  TERMINATE THE RUN
C
C/ INSERT COMBAS
C/ INSERT COMDDP
C/ INSERT COMDIM
C
C      CHECK IF A NORMAL TERMINATION
       IF(.NOT.NLEND) GO TO 100
          CALL BLINES(2)
         CALLMESAGE(48H   4.2 TERMINATE THE RUN                            )
          WRITE(18,20)
 20       FORMAT('+',50X,'*** NORMAL EXIT')
          CALL BLINES(2)
          CALL DAYTIM
          CALL RUNTIM
          CALL PAGE
          STOP
C
C      ERROR WAS FOUND SOMEWHERE. ABNORMAL TERMINATION
  100 CONTINUE
      CALL BLINES(3)
      CALLMESAGE(48H   4.2 ABNORMAL EXIT                                 )
           IF(NERROR.EQ.1)
     # CALLMESAGE(48H  ***  SOLUTION FAILED TO CONVERGE IN ELEPOT     )
           IF(NERROR.EQ.2)
     # CALLMESAGE(48H  ***  GEOMETRIC ERROR DETECTED IN ROUTINE QCALC)
           IF(NERROR.EQ.3)
     # CALLMESAGE(48H  ***  ERROR RETURN FROM ROUTINE TCALC           )
           IF(NERROR.EQ.4)
     # CALLMESAGE(48H  ***    ERROR RETURN FROM SUBROUTINE ELECOF     )
           CALL OUTGRD(4)
           CALL OUTGRD(5)
           CALL OUTGRD(3)
C
C     PRINT THE FINAL TIMES
         CALL BLINES(2)
         CALL DAYTIM
         CALL RUNTIM
         CALL PAGE
C
         STOP  2230
         END
C
C----------------------------------------------------------------
C
         SUBROUTINE MESAGE(KMESS)
C
C U.1  PRINT 48-CHARACTER MESSAGE ON OUTPUT CHANNEL
C
C     VERSION 1B           17/12/73      KVR/MHH        CULHAM
C
C/ INSERT COMBAS
       DIMENSION KMESS(12)
C
         WRITE(NOUT,9900) (KMESS(J),J=1,12)
C
         RETURN
 9900    FORMAT(4X,12A4)
         END
C
C----------------------------------------------------------------
C
         SUBROUTINE PAGE
C
C U.2  FETCH NEW PAGE ON OUTPUT CHANNEL
C
C     VERSION 1B           17/12/73      KVR/MHH        CULHAM
C
C/ INSERT COMBAS
         WRITE(NOUT,9900)
C
         RETURN
 9900    FORMAT(1H1)
         END
C
C----------------------------------------------------------------
C
         SUBROUTINE BLINES(K)
C
C U.3  INSERT BLANK LINES ON OUTPUT CHANNEL
C
C     VERSION 1B           17/12/73      KVR/MHH        CULHAM
C
C/ INSERT COMBAS
C
         DO 100 J=1,K
  100    WRITE(NOUT,9900)
C
         RETURN
 9900    FORMAT(1H )
         END
```

```
C
C----------------------------------------------------------------
C
        SUBROUTINE RUNTIM
C
C U.12 UPDATE CPU TIME (MINS) AND PRINT IT
C
C    VERSION 2B          17/12/73      KVR/MHH        CULHAM
C
C/ INSERT COMBAS
C
C    TIME(1,0,ICPUT) IS A MTS LIBRARY FUNCTION GIVING THE CPU-TIME
C    USED SO FAR (ICPUT IN MSECS)
          CALL TIME(1,0,ICPUT)
         ISEC = ICPUT/1000
         MIN = ISEC/60
         ISEC = ISEC - 60*MIN
         SEC = ISEC
         CPTIME = MIN + SEC/60.
        WRITE(NOUT,9900) MIN,ISEC
C
        RETURN
 9900   FORMAT(5X,22HCPU TIME USED SO FAR =,I4,5H MINS,I4,5H SECS)
        END
C
C--------------------------------------------------------------
C
        SUBROUTINE DAYTIM
C
C U.13 PRINT DATE AND TIME
C
C    VERSION 2B          17/12/73      KVR/MHH        CULHAM
C
C/ INSERT COMBAS
C    TIME IS AN MTS LIBRARY ROUTINE WHICH RETURNS THE DATE AND TIME
         DIMENSION DATE(4),TIIME(2)
         CALL TIME(21,0,DATE)
         CALL TIME(4,0,TIIME)
C
         WRITE(NOUT,9900) DATE,TIIME
        RETURN
 9900    FORMAT(5X,4A4,5X,2A4)
        END
C
C----------------------------------------------------------
C
        SUBROUTINE RESETR(PA,KDIM,PVALUE)
C
C U.14 RESET REAL ARRAY TO SPECIFIED VALUE
C
C    VERSION 1B          17/12/73      KVR/MHH        CULHAM
C
      DIMENSION   PA(KDIM)
C
        DO 100 J=1,KDIM
        PA(J) = PVALUE
  100   CONTINUE
C
        RETURN
        END
```

```
C
C-----------------------------------------------------------------
C
      SUBROUTINE RESETI(KA,KDIM,KVALUE)
C
C U.15 RESET INTEGER ARRAY TO SPECIFIED VALUE
C
C     VERSION 1B           17/12/73      KVR/MHH        CULHAM
C
      DIMENSION   KA(KDIM)
      DO 100 J=1,KDIM
      KA(J) = KVALUE
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE RESETH(KA,KDIM,KVALUE)
C
C U.16 RESET HOLLERITH ARRAY TO SPECIFIED VALUE
C
C     VERSION 1B           17/12/73      KVR/MHH        CULHAM
C
      DIMENSION   KA(KDIM), KVALUE(1)
C
      DO 100 J=1,KDIM
      KA(J) = KVALUE(1)
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE JOBTIM(PTIME)
C
C U.17 FETCH ALLOCATED JOBTIME (MINS)
C
C     VERSION 2B           17/12/73      KVR/MHH        CULHAM
C
C      SET CPU TIME COUNTER TO 0.
      CALL TIME(0,0,JUNK)
C
C     GUINFO(78,INFO1) IS AN MTS LIBRARY FUNCTION WHICH GIVES THE GLOBAL
C     TIME REMAINING
C     GUINFO(86,INFO2) IS AN MTS LIBRARY FUNCTION WHICH GIVES THE LOCAL
C     TIME REMAINING
C     INFO1 AND INFO2 ARE IN CPU TIME UNITS.  ONE TIME UNIT
C      EQUALS 13.0208333333 MICROSECS.  CPTIME IS IN MINS.
      CALL GUINFO(78,INFO1)
      CALL GUINFO(86,INFO2)
      INFO = MIN0(INFO1,INFO2)
      PTIME = 2.1701389*INFO
C
      RETURN
      END
C
C-----------------------------------------------------------------
```

```
C
         SUBROUTINE RESETL(KLA,KDIM,KLVAL)
C
C U.19 RESET LOGICAL ARRAY TO SPECIFIED VALUE
C
C     VERSION 1B            17/12/73      KVR/MHH        CULHAM
C
       LOGICAL     KLA,      KLVAL
       DIMENSION    KLA(KDIM)
         DO 100 J=1,KDIM
         KLA(J) = KLVAL
  100    CONTINUE
C
         RETURN
         END
C
C---------------------------------------------------------------
C
         SUBROUTINE SCALER(PA,KDIM,PC)
C
C U.21 SCALE A REAL ARRAY BY A REAL VALUE
C
C     VERSION 1B            17/12/73      KVR/MHH        CULHAM
C
       DIMENSION PA(KDIM)
C
         DO 100 J=1,KDIM
         PA(J) = PA(J) * PC
  100    CONTINUE
C
         RETURN
         END
C
C-----------------------------------------------------------
C
         SUBROUTINE COPYR(PA1,K1,PA2,K2,KDIM)
C
C U.23 COPY ONE REAL ARRAY INTO ANOTHER
C
C     VERSION 1B            17/12/73      KVR/MHH        CULHAM
C
       DIMENSION   PA1(KDIM),PA2(KDIM)
C
         DO 100 J=1,KDIM
         I1 = K1 + J - 1
         I2 = K2 + J - 1
         PA2(I2) = PA1(I1)
  100    CONTINUE
C
         RETURN
         END
END OF FILE
C/ MODULE COMBAS
C
C  1.1           BASIC SYSTEM PARAMETERS
      COMMON /COMBAS/
     R    ALTIME,     CPTIME,     STIME,
     I    LABEL1,     LABEL2,     LABEL3,    LABEL4,
     I    LABEL5,     LABEL6,     LABEL7,    LABEL8,
     I    NDIARY,        NIN,     NLEDGE,    NONLIN,
     I       NOUT,      NPRINT,      NPUNCH,      NREAD,
     I       NREC,      NRESUM,       NRUN,       NSTEP,
     L      NLEND,       NLRES
C
      LOGICAL NLEND,NLRES
C
      INTEGER*4  LABEL1(12),LABEL2(12),LABEL3(12),LABEL4(12),
     I           LABEL5(12),LABEL6(12),LABEL7(12),LABEL8(12)
C/ MODULE COMDDP
C
C  1.9          DEVELOPMENT AND DIAGNOSTIC PARAMETERS
      COMMON /COMDDP/
     I     MAXDUM,     MXDUMP,      NADUMP,      NCLASS,
     I     NPDUMP,     NPOINT,        NSUB,      NVDUMP,
     L     NLCHED,     NLHEAD,      NLOMT1,      NLOMT2,
     L     NLOMT3,     NLREPT
C
      LOGICAL  NLCHED,NLHEAD(9),NLOMT1(50),NLOMT2(50),NLOMT3(50),
     L         NLREPT
C
      INTEGER*4  NADUMP(20),NPDUMP(20),NVDUMP(20)
C/ MODULE COMGLO
C
C  1.2  GLOBAL VARIABLES
          COMMON /COMGLO/
     #          ELECON,QTHERM,POTENT,TEMP,
     #          DELTAX,DELTAY,THICK,DTIME,TIME,
     #          XCOORD,YCOORD,XMIN,XMAX,YMIN,YMAX,
     #          ELEGEO,THMGEO
C
          REAL ELECON(50,50),QTHERM(50,50),DELTAX(50),
     #          POTENT(50,50),TEMP(50,50),DELTAY(50),
     #          XCOORD(50),YCOORD(50)
          INTEGER*4 ELEGEO(50,50),THMGEO(50,50)
C/ MODULE COMELE
C
C  1.3  ELECTRICAL CONDUCTIVITY AND DIFFERENCE EQU. COEFFICENTS
          COMMON /COMELE/
     #          ECXP,ECXM,ECYP,ECYM,ERHS,EXMXP,EYMYP,
     #          ELEVOL,ELEALP,ELEBET,ELEPAR,EPSELE
C
          REAL ECXP(50,50),ECXM(50,50),ECYP(50,50),ECYM(50,50),
     #          ELEVOL(10),ELEALP(10),ELEBET(10),ELEPAR(50),
     #          ERHS(50,50),EXMXP(50,50),EYMYP(50,50)
C/ MODULE COMTHM
C
C  1.4  THERMAL CONDUCTIVITY AND HEAT CAPACITY
          COMMON /COMTHM/
     #          THMCON,THMCAP,THMTEM,TCXP,TCXM,TCYP,TCYM,
     #          TCX,TCY,TCRHS
C
          REAL THMCON(10),THMCAP(10),THMTEM(10)
          REAL TCXP(50,50),TCXM(50,50),TCYP(50,50),TCYM(50,50)
          REAL TCX(50,50),TCY(50,50),TCRHS(50,50)
C/ MODULE COMTEM
C
C  1.5  TEMPERARY ARRAYS
          COMMON /COMTEM/
     #          TEM1,TEM2,TEM3,T1,T2,T3,T4,T5
C
```

```
                REAL TEM1(50,50),TEM2(50,50),TEM3(50,50),T1(50),
     #               T2(50),T3(50),T4(50),T5(50)
C/ MODULE COMDIM
C
C  1.6  DIMENSION AND COUNTER VARIABLES
                 COMMON /COMDIM/
     #           NX,NY,NEQU,NALPHA,MITER,MMAX,NREG,NGEO,NERROR,NTYPE,
     #           NSPO,NPX,NPY,NIJ,NCI,NDX,NDY,NSTO
C
                 INTEGER*4 NX,NY,NDX,NDY,NEQU,NALPHA,MITER,MMAX
                 INTEGER*4 NREG,NGEO,NERROR,NTYPE,NSPO,NPX,NPY
                 INTEGER*4 NIJ,NCI,NSTO
C/ MODULE COMCON
C
C  1.7  CONTROL THE HEATING
                  COMMON /COMCON/
     #                  HTIME,CTIME,CCUR,CPOW,DTEMP,DELT,CUR,
     #                  VOLTS,RESIST,POWER,CVOL,TINIT,DEENG,TEENG,
     #                  DQENG,QENG,TQENG,TTENG,TENG,SCALE
C
                  REAL QENG(10),TENG(10)
END OF FILE
```

Appendix II - Sample Output from MEGAERA

```
&INMOD NRUN=100 &END
&LABELS
  LABEL1='MEGAERA RUN NO. 35    PARALLEL ELECTRODES',
  LABEL2='OVERBURDEN CONDUCTIVITY RATIO   OS/OB = 1.0',
  LABEL4='GEOMETRY SET UP AND ENTERED BY ALLAN HIEBERT',
&END
&INPUT1     NX=50,NY=50,
  DELTAX=50*1.5,DELTAY=8*5.0,2.5,3*2.0,1.5,25*1.2,
  1.5,3*2.0,2.5,6*5.0,10.0,NREG=5,
&END
&REGION   MINI=1,MAXI=50,MINJ=14,MAXJ=38,
   ETYPE='DOMA',EALPHA=2.29E-2,EBETA=1.E-3,
   TTYPE='COND',THMRC=1.6E6,THMK=1.8,
&END
&REGION   MINI=1,MAXI=50,MINJ=1,MAXJ=13,
   ETYPE='DOMA',EALPHA=2.29E-2,EBETA=1.0E-3,
   TTYPE='COND',THMRC=1.6E6,THMK=1.8,
&END
&REGION   MINI=1,MAXI=50,MINJ=39,MAXJ=50,
   ETYPE='DOMA',EALPHA=2.29E-2,EBETA=1.0E-3,
   TTYPE='COND',THMRC=1.6E6,THMK=1.8,
&END
&REGION   MINI=1,MAXI=1,MINJ=24,MAXJ=29,
   ETYPE='ELEC',VOLTS=1000.,
   TTYPE='COND',THMRC=1.E6,THMK=3.0,
&END
&REGION   MINI=50,MAXI=50,MINJ=24,MAXJ=29,
   ETYPE='ELEC',VOLTS=0.,
   TTYPE='COND',THMRC=1.E6,THMK=3.0,
&END
&INPUT2
   NSPO=10,XMIN=1.0,XMAX=49.0,NPX=17,
   YMIN=40,YMAX=85,NPY=10,
&END
&INPUT3
   TINIT=15.,DTIME=1.E5,HTIME=3.154E7,NTYPE=3,
   CPOW=12.E3,CVOL=1000.,DTEMP=.05,NCI=20,
   MMAX=350,EPSELE=.02,
&END
&INPLOT
   LABEL1='PARALLEL PLATE RUN NO. 35     80/11/25',NLAB1=38,
   LABEL2='CONDUCTIVITY RATIO    OB/OS = 1.0/ 1.0',NLAB2=38,
   NCPLTS=1,CPTIME=3.15E7,NCCODE=3,
   CXMIN=1.0,CXMAX=74.0,CYMIN=30.,CYMAX=100.,
   NPPLTS=1,PPTIME=3.15E7,NPCODE=7,
   PXMIN=1.0,PXMAX=99.0,PYMIN=15.,PYMAX=115.,
&END
```

```
                         P R O G R A M   M E G A E R A
                         -----------------------------

                         ADH              AUGUST, 1980

MEGAERA RUN NO. 35    PARALLEL ELECTRODES
OVERBURDEN CONDUCTIVITY RATIO   OS/OB = 1.0

GEOMETRY SET UP AND ENTERED BY ALLAN HIEBERT


THICKNESS OF 2D SLICE (THICK)      =  1.0000E+00
GEOMETRY OF PROBLEM                =        VERT
NUMBER OF REGIONS (NREG)           =           5


REGION NUMBER                      =           1
ELECTRICAL TYPE                    =        DOMA
TEMP. DEPENDENCE OF CONDUCTIVITY   =  2.2900E-02
CONDUCTIVITY AT 24 CELSIUS         =  1.0000E-03
THERMAL REGION TYPE                =        COND
THERMAL CONDUCTIVITY               =  1.8000E+00
THERMAL HEAT CAPACITY              =  1.6000E+06


REGION NUMBER                      =           2
ELECTRICAL TYPE                    =        DOMA
TEMP. DEPENDENCE OF CONDUCTIVITY   =  2.2900E-02
CONDUCTIVITY AT 24 CELSIUS         =  1.0000E-03
THERMAL REGION TYPE                =        COND
THERMAL CONDUCTIVITY               =  1.8000E+00
THERMAL HEAT CAPACITY              =  1.6000E+06


REGION NUMBER                      =           3
ELECTRICAL TYPE                    =        DOMA
TEMP. DEPENDENCE OF CONDUCTIVITY   =  2.2900E-02
CONDUCTIVITY AT 24 CELSIUS         =  1.0000E-03
THERMAL REGION TYPE                =        COND
THERMAL CONDUCTIVITY               =  1.8000E+00
THERMAL HEAT CAPACITY              =  1.6000E+06


REGION NUMBER                      =           4
ELECTRICAL TYPE                    =        ELEC
VOLTAGE ( BEFORE SCALING )         =  1.0000E+03
THERMAL REGION TYPE                =        COND
THERMAL CONDUCTIVITY               =  3.0000E+00
THERMAL HEAT CAPACITY              =  1.0000E+06


REGION NUMBER                      =           5
ELECTRICAL TYPE                    =        ELEC
VOLTAGE ( BEFORE SCALING )         =  0.0
```

```
THERMAL REGION TYPE                =        COND
THERMAL CONDUCTIVITY               =  3.0000E+00
THERMAL HEAT CAPACITY              =  1.0000E+06


NO. OF STEPS PER PRINTED OUTPUT    =          10
NO. OF STEPS PER TAPE STORAGE      =        1000
NO. OF INTERPOLATION POINTS IN X   =          17
NO. OF INTERPOLATION POINTS IN Y   =          10
X MINIMUM FOR INTERPOLATION        =  1.0000E+00
X MAXIMUM FOR INTERPOLATION        =  4.9000E+01
Y MINIMUM FOR INTERPOLATION        =  4.0000E+01
Y MAXIMUM FOR INTERPOLATION        =  8.5000E+01



INITIAL TEMPERATURE IN CELSIUS     =  1.5000E+01
TOTAL HEATING TIME IN SECONDS      =  3.1540E+07
INITIAL TIMESTEP SIZE IN SECONDS   =  1.0000E+05
GRIDLINE FOR CURRENT INTEGRAL      =          20
CONSTANT POWER . ( WATTS )         =  1.2000E+04
CHANGE IN TEMPERATURE / TIMESTEP   =  5.0000E-02
MAXIMUM NO. OF ITERATIONS (MMAX)   =         350
ELECTRICAL CONVERGENCE CRITERIA    =  2.0000E-02
NUMBER OF ITERATION PARAMETERS     =          50
```

```
          ELECTRICAL REGIONS OF THE GRID (NOT TO SCALE)
  ROW
   50     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   49     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   48     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   47     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   46     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   45     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   44     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   43     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   42     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   41     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   40     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   39     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
   38     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   37     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   36     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   35     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   34     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   33     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   32     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   31     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   30     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   29      4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1 5
   28      4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1 5
   27      4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1 5
   26      4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1 5
   25      4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1 5
   24      4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1 5
   23     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   22     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   21     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   20     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   19     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   18     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   17     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   16     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   15     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   14     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
   13     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
   12     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
   11     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
   10     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
    9     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
    8     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
    7     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
    6     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
    5     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
    4     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
    3     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
    2     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
    1     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2

 COLUMN    1 2 3 4 5 6 7 8 91011121314151617181920212223242526272829303132333435363738394041424344454647484950
```

```
THERMAL REGIONS OF THE GRID  (NOT TO SCALE)

ROW
50     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
49     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
48     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
47     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
46     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
45     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
44     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
43     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
42     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
41     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
40     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
39     -3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3
38     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
37     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
36     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
35     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
34     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
33     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
32     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
31     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
30     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
29     -4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-5
28     -4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-5
27     -4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-5
26     -4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-5
25     -4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-5
24     -4-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-5
23     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
22     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
21     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
20     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
19     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
18     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
17     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
16     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
15     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
14     -1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
13     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
12     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
11     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
10     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
 9     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
 8     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
 7     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
 6     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
 5     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
 4     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
 3     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
 2     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
 1     -2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2

COLUMN  1 2 3 4 5 6 7 8 91011121314151617181920212223242526272829303132333435363738394041424344454647484950
```

```
TIMESTEP NUMBER                    =          1

TIME AT END OF STEP                =  1.3055E+04
SIZE OF TIMESTEP                   =  1.3055E+04

NO. OF ITERATIONS OF A.D.I.P.      =         226
VOLTAGE BETWEEN ELECTRODES         =  5.2318E+03
POWER INPUT TO DOMAIN              =  1.2000E+04
CURRENT BETWEEN ELECTRODES         =  2.2937E+00
RESISTANCE BETWEEN ELECTRODES      =  2.2810E+03

ENERGY BALANCE       ( JOULES )
TIMESTEP  :  ELECTRICAL ENERGY =  1.5666E+08     HEAT PRODUCED =  1.5809E+08
CUMULATIVE:  ELECTRICAL ENERGY =  1.5666E+08     HEAT GENERATED =  1.5809E+08     TEMP. RISE ENERGY =  1.5801E+08
    REGION =  1     HEAT GENERATED =  1.2496E+08      TEMP. RISE ENERGY =  1.2484E+08
    REGION =  2     HEAT GENERATED =  1.5665E+07      TEMP. RISE ENERGY =  1.5634E+07
    REGION =  3     HEAT GENERATED =  1.7467E+07      TEMP. RISE ENERGY =  1.7437E+07
    REGION =  4     HEAT GENERATED =  0.0             TEMP. RISE ENERGY =  4.8345E+04
    REGION =  5     HEAT GENERATED =  0.0             TEMP. RISE ENERGY =  5.4437E+04


     ELECTRICAL POTENTIAL IN VOLTS

    X (M)     1.000     4.000     7.000    10.000    13.000    16.000    19.000    22.000    25.000    28.000
 Y (M)
85.000       3578.0    3557.6    3515.4    3456.1    3384.8    3305.7    3222.0    3135.6    3047.9    2959.6
80.000       3826.4    3790.1    3719.3    3627.0    3524.0    3417.0    3309.5    3203.2    3098.7    2995.9
75.000       4207.6    4121.0    3979.5    3824.1    3671.7    3527.6    3392.3    3264.8    3143.7    3027.5
70.000       4963.0    4578.8    4259.0    4004.3    3793.2    3612.3    3452.6    3308.2    3174.7    3048.9
65.000       5207.7    4780.2    4370.0    4069.4    3834.3    3639.6    3471.6    3321.6    3184.2    3055.5
60.000       4709.5    4463.1    4197.7    3967.5    3769.5    3596.3    3441.6    3300.5    3169.4    3045.5
55.000       4097.4    4030.0    3912.3    3775.8    3637.1    3502.5    3374.1    3251.8    3134.6    3021.6
50.000       3757.9    3726.9    3665.3    3583.1    3489.3    3390.0    3288.9    3187.9    3087.7    2988.7
45.000       3529.8    3511.6    3473.7    3419.8    3354.3    3280.8    3202.1    3120.4    3036.9    2952.3
40.000       3365.9    3353.7    3327.6    3289.4    3241.2    3185.3    3123.5    3057.4    2988.2    2917.0

    X (M)    31.000    34.000    37.000    40.000    43.000    46.000    49.000
 Y (M)
85.000       2871.1    2782.4    2693.5    2604.4    2514.9    2425.0    2334.9
80.000       2894.6    2794.1    2694.1    2593.7    2492.6    2390.0    2285.5
75.000       2914.8    2804.3    2694.7    2584.9    2473.7    2359.9    2242.2
70.000       2928.4    2811.1    2695.2    2579.2    2461.3    2339.8    2212.8
65.000       2932.6    2813.3    2695.6    2577.8    2457.9    2334.2    2204.4
60.000       2926.5    2810.5    2695.9    2581.1    2464.5    2344.5    2219.3
55.000       2911.5    2803.4    2696.0    2588.5    2479.6    2368.4    2253.8
50.000       2890.6    2793.2    2696.1    2598.6    2500.5    2401.2    2300.3
45.000       2867.2    2781.8    2696.1    2610.2    2524.0    2437.6    2351.3
40.000       2844.3    2770.6    2696.4    2621.9    2547.6    2473.7    2400.7

     HEATING RATE IN JOULES/SECOND-METER**3

    X (M)     1.000     4.000     7.000    10.000    13.000    16.000    19.000    22.000    25.000    28.000
 Y (M)
85.000          1.4       1.3       1.3       1.1       1.0       1.0       0.9       0.8       0.8       0.7
80.000          3.0       2.7       2.3       2.0       1.7       1.4       1.3       1.1       1.0       1.0
75.000          8.2       6.4       4.6       3.4       2.6       2.1       1.7       1.5       1.3       1.2
70.000         67.1      18.3       8.9       5.3       3.6       2.6       2.1       1.7       1.5       1.3
65.000          4.7      20.9      10.8       6.2       4.0       2.9       2.2       1.8       1.6       1.4
60.000         32.4      14.1       7.8       4.9       3.4       2.5       2.0       1.7       1.4       1.3
```

```
55.000           6.1      5.1      3.9      3.0      2.3      1.9      1.6      1.4      1.2      1.1
50.000           2.4      2.3      2.0      1.7      1.5      1.3      1.1      1.0      1.0      0.9
45.000           1.2      1.2      1.1      1.0      0.9      0.9      0.8      0.7      0.7      0.7
40.000           0.7      0.7      0.6      0.6      0.6      0.6      0.5      0.5      0.5      0.5

     X (M)    31.000   34.000   37.000   40.000   43.000   46.000   49.000
  Y (M)
85.000           0.7      0.7      0.7      0.7      0.7      0.8      0.8
80.000           0.9      0.9      0.9      0.9      0.9      1.0      1.1
75.000           1.1      1.1      1.1      1.1      1.1      1.2      1.3
70.000           1.2      1.2      1.2      1.2      1.3      1.4      1.5
65.000           1.3      1.2      1.2      1.2      1.3      1.4      1.6
60.000           1.2      1.2      1.2      1.2      1.2      1.3      1.5
55.000           1.1      1.0      1.0      1.0      1.1      1.1      1.3
50.000           0.9      0.8      0.8      0.8      0.9      0.9      1.0
45.000           0.7      0.7      0.7      0.7      0.7      0.7      0.7
40.000           0.5      0.5      0.5      0.5      0.5      0.5      0.5

     TEMPERATURE IN DEGREES CELSIUS

     X (M)     1.000    4.000    7.000   10.000   13.000   16.000   19.000   22.000   25.000   28.000
  Y (M)
85.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0
80.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0
75.000          15.1     15.1     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0
70.000          15.5     15.1     15.1     15.0     15.0     15.0     15.0     15.0     15.0     15.0
65.000          15.0     15.2     15.1     15.1     15.0     15.0     15.0     15.0     15.0     15.0
60.000          15.3     15.1     15.1     15.0     15.0     15.0     15.0     15.0     15.0     15.0
55.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0
50.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0
45.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0
40.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0     15.0

     X (M)    31.000   34.000   37.000   40.000   43.000   46.000   49.000
  Y (M)
85.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0
80.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0
75.000          15.0     15.0     15.0     15.0     15.0     15.0      5.0
70.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0
65.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0
60.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0
55.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0
50.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0
45.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0
40.000          15.0     15.0     15.0     15.0     15.0     15.0     15.0
```

```
 TIMESTEP NUMBER                    =          79

 TIME AT END OF STEP                =  3.1540E+07
 SIZE OF TIMESTEP                   =  1.5970E+06

 NO. OF ITERATIONS OF A.D.I.P.      =          18
 VOLTAGE BETWEEN ELECTRODES         =  3.0009E+03
 POWER INPUT TO DOMAIN              =  1.2000E+04
 CURRENT BETWEEN ELECTRODES         =  3.9988E+00
 RESISTANCE BETWEEN ELECTRODES      =  7.5046E+02

 ENERGY BALANCE       ( JOULES )
 TIMESTEP   :  ELECTRICAL ENERGY =  1.9164E+10     HEAT PRODUCED =  1.9173E+10
 CUMULATIVE:   ELECTRICAL ENERGY =  3.7848E+11     HEAT GENERATED =  3.7824E+11     TEMP. RISE ENERGY =  3.7819E+11
     REGION =  1     HEAT GENERATED =  2.6882E+11       TEMP. RISE ENERGY =  2.5076E+11
     REGION =  2     HEAT GENERATED =  5.1464E+10       TEMP. RISE ENERGY =  5.8573E+10
     REGION =  3     HEAT GENERATED =  5.6879E+10       TEMP. RISE ENERGY =  6.4705E+10
     REGION =  4     HEAT GENERATED =  0.0              TEMP. RISE ENERGY =  2.0873E+09
     REGION =  5     HEAT GENERATED =  0.0              TEMP. RISE ENERGY =  2.0876E+09

         ELECTRICAL POTENTIAL IN VOLTS

      X (M)     1.000     4.000     7.000    10.000    13.000    16.000    19.000    22.000    25.000    28.000
  Y (M)
 85.000        2382.3    2366.1    2331.8    2281.9    2219.3    2146.9    2067.1    1981.8    1892.6    1800.7
 80.000        2529.3    2506.8    2460.7    2396.4    2348.7    2231.4    2137.6    2039.1    1937.7    1834.2
 75.000        2695.4    2657.6    2589.3    2503.8    2407.7    2304.7    2197.2    2086.8    1974.7    1861.5
 70.000        2927.9    2812.6    2697.5    2584.5    2470.4    2354.5    2236.8    2117.9    1998.6    1879.0
 65.000        2994.1    2870.4    2734.4    2610.3    2489.8    2369.4    2248.5    2127.1    2005.5    1884.1
 60.000        2857.1    2777.0    2675.5    2569.0    2458.7    2345.3    2229.5    2112.3    1994.2    1875.8
 55.000        2652.4    2620.0    2558.8    2479.2    2387.8    2288.6    2184.3    2076.5    1966.7    1855.6
 50.000        2491.9    2471.4    2428.8    2368.5    2294.8    2211.3    2121.0    2025.7    1927.2    1826.4
 45.000        2347.9    2332.7    2300.6    2253.6    2194.3    2125.3    2048.9    1966.9    1880.8    1791.9
 40.000        2218.9    2207.2    2182.1    2144.7    2096.8    2040.1    1976.2    1906.7    1832.9    1755.8

      X (M)    31.000    34.000    37.000    40.000    43.000    46.000    49.000
  Y (M)
 85.000        1706.9    1612.0    1516.4    1420.8    1325.6    1231.3    1138.7
 80.000        1729.4    1624.0    1518.1    1412.2    1306.6    1201.5    1097.6
 75.000        1747.7    1633.7    1519.5    1405.3    1291.2    1177.3    1063.9
 70.000        1759.4    1639.8    1520.4    1400.9    1281.4    1161.8    1042.2
 65.000        1762.7    1641.6    1520.6    1399.6    1278.5    1157.3    1035.9
 60.000        1757.3    1638.7    1520.2    1401.7    1283.1    1164.6    1046.1
 55.000        1743.8    1631.6    1519.2    1406.8    1294.5    1182.5    1071.1
 50.000        1724.2    1621.2    1517.7    1414.2    1310.9    1208.4    1107.1
 45.000        1700.9    1608.8    1516.0    1423.0    1330.6    1239.2    1149.5
 40.000        1676.5    1595.7    1514.1    1432.4    1351.3    1271.4    1193.5

         HEATING RATE IN JOULES/SECOND-METER**3

      X (M)     1.000     4.000     7.000    10.000    13.000    16.000    19.000    22.000    25.000    28.000
  Y (M)
 85.000           1.3       1.3       1.3       1.3       1.4       1.4       1.4       1.4       1.4       1.3
 80.000           2.3       2.2       2.2       2.1       2.1       2.1       2.0       2.0       1.9       1.9
 75.000           5.0       4.3       3.8       3.4       3.1       2.9       2.8       2.6       2.5       2.5
 70.000          32.9      10.4       6.4       5.0       4.2       3.8       3.4       3.2       3.0       2.9
 65.000           2.4      11.6       7.5       5.6       4.6       4.1       3.7       3.4       3.2       3.0
 60.000          16.4       8.3       5.8       4.6       4.0       3.6       3.3       3.1       2.9       2.8
```

```
55.000          4.0      3.6      3.3      3.1      2.9      2.7      2.6      2.5      2.4      2.3
50.000          2.0      1.9      1.9      1.9      1.9      1.9      1.8      1.8      1.8      1.7
45.000          1.2      1.2      1.2      1.2      1.2      1.2      1.2      1.2      1.2      1.2
40.000          0.7      0.7      0.8      0.8      0.8      0.8      0.8      0.8      0.8      0.8

      X (M)  31.000   34.000   37.000   40.000   43.000   46.000   49.000
  Y (M)
85.000          1.3      1.3      1.3      1.3      1.3      1.3      1.4
80.000          1.9      1.8      1.8      1.8      1.8      1.9      1.9
75.000          2.4      2.4      2.3      2.3      2.4      2.4      1.9
70.000          2.8      2.8      2.7      2.7      2.8      2.9      2.5
65.000          2.9      2.9      2.9      2.9      2.9      3.0      3.0
60.000          2.7      2.7      2.7      2.7      2.7      2.8      3.1
55.000          2.3      2.2      2.2      2.2      2.3      2.3      2.9
50.000          1.7      1.7      1.7      1.7      1.7      1.7      2.4
45.000          1.2      1.2      1.2      1.2      1.2      1.2      1.8
40.000          0.9      0.9      0.9      0.9      0.9      0.9      1.2
                                                                     0.8
      TEMPERATURE  IN DEGREES CELSIUS

      X (M)   1.000    4.000    7.000   10.000   13.000   16.000   19.000   22.000   25.000   28.000
  Y (M)
85.000         54.0     53.1     51.4     49.2     47.1     45.1     43.4
80.000         83.7     80.7     75.5     69.5     63.9     59.1     55.2     42.0     40.9     40.1
75.000        136.3    126.6    111.8     97.3     85.0     75.5     68.3     52.2     50.0     48.3
70.000        209.2    182.4    150.9    124.4    104.1     89.4     78.9     63.0     59.2     56.5
65.000        207.5    197.0    164.4    134.0    110.8     94.1     82.3     71.4     66.2     62.5
60.000        194.0    170.7    143.0    119.0    109.4     86.7     76.8     74.1     68.4     64.4
55.000        121.5    114.1    102.4     90.3     79.9     71.6     65.3     69.8     64.8     61.4
50.000         75.3     73.0     68.9     64.1     59.5     55.5     52.2     60.5     57.0     54.6
45.000         49.4     48.7     47.4     45.7     44.0     42.4     41.0     49.7     47.7     46.2
40.000         35.4     35.2     34.8     34.3     33.7     33.2     32.7     39.8     38.9     38.2
                                                                           32.3     31.9     31.6
      X (M)  31.000   34.000   37.000   40.000   43.000   46.000   49.000
  Y (M)
85.000         39.5     39.1     39.0     39.0     39.3     39.8     40.6
80.000         47.1     46.4     46.2     46.3     46.8     47.8     49.3
75.000         54.6     53.5     53.1     53.3     54.2     55.8     58.2
70.000         60.1     58.7     58.1     58.4     59.5     61.6     64.8
65.000         61.8     60.3     59.7     60.0     61.2     63.4     66.9
60.000         59.1     57.7     57.2     57.4     58.5     60.5     63.5
55.000         52.9     51.9     51.5     51.7     52.5     53.9     56.1
50.000         45.2     44.6     44.3     44.4     44.9     45.8     47.1
45.000         37.7     37.4     37.2     37.3     37.5     38.0     38.6
40.000         31.4     31.3     31.3     31.3     31.4     31.6     31.8

    4.2 TERMINATE THE RUN

  MON DEC 15/80        09:27:08
  CPU TIME USED SO FAR =   2 MINS  37 SECS
```